(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number : **0 689 207 A1**

(12) **EUROPEAN PATENT APPLICATION**

(21) Application number : 95201658.2

(22) Date of filing : 19.06.95

(51) Int. Cl.^{6} : **G11B 27/00, G06F 3/06**

(72) Inventor : Verinky, Phil
1425 Lupton Avenue
San Jose, California 95125 (US)
Inventor : Case, Michael
825 E. Evelyn Avenue,
Apt. 451
Sunnyvale, California 94086 (US)

(74) Representative : Hall, Robert Leonard et al
Dibb, Lupton, Broomhead,
Fountain Precinct,
Balm Green
Sheffield S1 1RZ (GB)

(54) IDE/ATA CD drive controller

(57) A compact disk drive controller to control the access of information from an optical compact disk (CD) digital data storage device by a host computer using an integrated drive electronics (IDE) data bus or an industry standard architecture (ISA) data bus is disclosed. A digital signal processor (DSP) interface to the drive electronics of the CD drive, a dynamic random access memory (DRAM) controller, an error correction code (ECC) data corrector, an error detection and correction (EDC) device employing cyclical redundancy checking techniques (EDC/CRC), and a host computer interface are described.

FIG. 2

EP 0 689 207 A1

BACKGROUND OF THE INVENTION

Field of the Invention

This invention relates generally to the access of digital data from optical storage media by a personal computer. Optical storage methods allow information to be recorded and recovered from a given material by using light. The compact disk (CD) media currently used in optical recording is capable of significantly higher areal density than magnetic disks. This capacity to store a large amount of information per unit area of the media surface is a major advantage of CD technology over magnetic disk storage techniques.

Prior Art

The field reliability of CD systems is yet to be determined due to the relatively short period of time this media has been in use. However, optical recording systems are expected to be more reliable than magnetic disk drives, generally referred to as hard disks, for several reasons. The optical heads used for recording and recovering information are spaced away from the disk at all times, eliminating the possibility of head crashes. And the optical techniques used cause no wear or tear on the media surface during reading or transferring of information.

The reliability characteristic of optical storage media appears to be especially advantageous where the removability and transportability of the media is critical. Compared to magnetic disk drives, both hard and floppy, the operation of a CD is much less sensitive or affected by dust accumulation on either the head or the media. And the optical methods of reading and writing data without making physical contact with the media surface significantly reduces the potential for damage in removable disk applications.

The integration of CD drives into personal computers comprises one of the largest markets for optical storage media applications for the foreseeable future. At present, the cost of a CD drive is a primary barrier to the growth of this market. However, the CD-ROM (read only memory) standard as originally developed by Sony and Phillips has become the standard defining the physical characteristics and disk format for data storage and retrieval. This format has become very popular for making large amounts of information available to users at a relatively low cost and there is an increasingly large library of CD-ROM titles available. CD drives which are capable of writing information to the CD are much less widely used today due to their much greater cost and complexity.

All CD drive designs include a CD load mechanism, a spindle, drive electronics and a controller. The drive electronics recovers data from the CD as directed by the controller. The controller manages the flow of commands, status flags and data between the host personal computer and the CD drive electronics.

Conventional CD drive designs support the Industry Standard Architecture (ISA) bus convention and require the insertion of an interface card or host adapter card into an ISA input/output bus slot of the host personal computer. These disk drive designs include a variety of proprietary and manufacturer specific designs as well as designs that support the three varying software driver formats used with the Small Computer Systems Interface (SCSI) standard. These three software driver formats include Microsoft's Layered Device Driver Architecture, the American National Standards Institutes' Common Access Method, and the Advanced SCSI Programming Interface.

A SCSI disk drive includes a controller and a SCSI slave interface. A SCSI disk drive communicates with a host computer through a SCSI host adapter card which must be resident on the ISA bus of the host. There are three types of host adapter cards, namely a register compatible controller, an INT 13h compatible controller, and an installable device driver. These types of host adapter designs are fully explained in widely available technical publications.

The reliance of all conventional CD drive designs exclusively on the use of the ISA input/output bus results in the additional expense of host adapter card electronics. Furthermore, a reduction in the range of employment of any given computer system due to the permanent commitment of an input/output bus slot to communication with the CD drive controller is a limitation in the prior art.

An alternative bus structure is available within standard personal computer architecture available for use with a CD drive controller. This structure is referred to as integrated drive electronics with an AT attachment interface, or IDE/ATA. The American National Standards Institute has published this standard and it is currently widely available. The term integrated drive electronics includes any drive with a controller included. For example, all SCSI drives are in fact IDE drives. The term IDE/ATA applies to a drive if and only if its interface conforms to the industry standard AT attachment specification. IDE/ATA drives do not take up an ISA input/output slot. This class of interface is connected by means of a dedicated 40 pin connector found on many personal computer mother boards.

Conventional CD drives in the prior art failed to make use of the IDE/ATA bus. However, now that the AT standard has become widely used in many personal computers, it would be desirable to provide a CD drive with built-in controller functionality and a standard connector. This would obviate the need for an additional host adapter card and associated electronics. Providing these electronics in addition to the CD drive itself increases the overall cost of a system using a CD drive for data storage and retrieval and also makes a CD drive more complicated to install on ex-

isting personal computers in use today. Due to the plethora of methods of ISA interface designs used in the industry today, compatibility issues often occur when, for example, a particular CD drive controller is tasked with communicating with another ISA bus connected peripheral device. The high frequency of incompatibility often prohibits the employment of the most cost efficient or highest performance combination of devices. The present invention, a controller for CD drives which can be implemented with a drive using a standard AT connector, overcomes the problems associated with the prior art as will be made clear in the following discussions thereof.

SUMMARY OF THE INVENTION

This invention relates to a compact disk drive controller for a compact disk drive to control the communication of digital information between a compact disk to a host computer. The compact disk drive would generally have it's own drive electronics comprising a digital signal processor, a microcontroller, a random access memory, a system controller and an ISA data bus. The host computer communicates with the compact disk drive controller via an IDE data bus and receives digital information from the compact disk via the IDE data bus. The compact disk drive controller is comprised of a host interface, connecting the host computer via the IDE data bus with the compact disk drive controller, in order to receive data addresses and commands from the host computer and transmit digital information to the host computer. A path for communicating data addresses and commands from the host interface to the microcontroller of the drive electronics is employed and a digital signal processor (DSP) interface connecting the host interface and the digital signal processor of the drive electronics, receives digital information from the compact disk and transmits the digital information to said host interface.

The digital signal processor interface of the compact disk drive controller (CDDC) further comprises a descrambler to descramble and assemble the digital information received from said digital signal processor and store said digital information into said random access memory.

The digital signal processor interface of the CDDC further comprises an error correction code circuit to perform error correction on said digital information. That error correction circuit could employ Reed-Solomon codes.

The digital signal processor interface of the CDDC further comprises a cyclic redundancy checker for detecting errors in the digital information after correction of the digital information by the error correction code circuit.

The host interface of the CDDC receives data addresses and commands from the host computer via

an ISA data bus and communicates digital information to the host computer via the ISA bus.

The host interface of the CDDC further comprises a command FIFO to transfer commands from the host computer to the system controller of the drive electronics of the compact disk drive.

The host interface of the CDDC further comprises a status FIFO to transfer status information from the drive electronics of the compact disk drive to the host computer.

The host interface of the CDDC further comprises a configuration register via which the host computer instructs the compact disk drive controller to present the digital information onto one of the ISA and IDE data buses in a data format selected from a group including 16-bit DMA, 8-bit DMA, 16-bit PIO, and an 8-bit PIO format.

A BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram of a CD drive configuration of the prior art with the IDE/ATA CD drive controller of the present invention added thereto.

FIG. 2 is a block diagram of an implementation of the present invention.

FIGS. 3a-3c are a pin diagram and accompanying pin-out assignments for an implementation of the present invention.

FIG. 4 is a pin description of the system controller interface of an implementation of this invention.

FIG. 5a is a pin description of the host interface of an implementation of this invention.

FIG. 5b is an address map of the host registers of an implementation of this invention.

FIG. 6 is a pin description of the DSP interface of an implementation of this invention.

FIG. 7 is a pin description of the subcode interface of an implementation of this invention.

FIG. 8a is a pin description of the RAM interface of an implementation of this invention.

FIG. 8b is a pin description for the RAM address bus of the RAM interface of an implementation of this invention.

FIG. 9 is a pin description of miscellaneous pins of an implementation of this invention.

FIG. 10 is a description of the address register.

FIG. 11 is a description of command packet register.

FIG. 12 is a description of interface status and interface control registers.

FIG. 13 and FIG. 14 are descriptions of DBCL and DBCH (Data Byte/Word Counter).

FIG. 15, FIG. 16, FIG. 17 and FIG. 18 are descriptions of HEAD0 to HEAD3 (Header Registers).

FIG. 19 is a list of DACH, DACL settings for various starting points.

FIG. 20 and FIG. 21 are ECC block pointer/write address counters.

DESCRIPTION OF A PREFERRED EMBODIMENT

Reference is now made to FIG. 1 which is a block diagram of a compact disk (CD) drive configuration of the prior art with a CD drive controller of the present invention added thereto. The CD drive controller designed according to this invention would communicate command data, status signals and other data over the integrated device electronics/AT attachment (IDE/ATA) bus of a personal computer. This invention reduces the cost of a CD drive by eliminating the need for a host adapter card or additional ISA bus interface electronics. This invention also allows the CD drive to integrate into many personal computers without requiring the use of an ISA input/output bus slot. Furthermore, this invention will allow for a wider selection of personal computer peripheral cards, such as sound and boards, for use with a given personal computer and CD drive. The method of the current invention reduces this potential for incompatibility, and permits a broad range of selection of peripheral devices.

The drive controller 10 is connected to drive electronics 12 of a CD drive 14 and IDE/ATA bus 16 of a personal computer. The clock speed of the controller will be determined by crystal oscillator 11. It will be understood that the compact disk 18 is not a part of the present invention but it is shown for clarity. The CD drive includes a spindle motor 20 for rotating the

CD and an optical head 22 for reading data from the CD. The drive electronics of the CD drive include a preamplifier 24 which sends a signal to servo control 26 of the CD drive for focus and tracking control. The servo control communicates with the spindle motor and optical head to position the optical head precisely to read the correct information from the CD. Digital data read from the preamplifier goes to digital signal processor (DSP) 28 in the drive electronics. The DSP sends subcode information as well as digital data to the drive controller of this invention. A microcontroller 29 in the CD drive electronics also communicates with the DSP and servo control of the drive electronics, as well as with the drive controller of the present invention, to control the reading of information from the CD. A DRAM 30 is coupled with the drive controller of the present invention for storing and buffering data via the drive controller. Data can be sent to digital-to-analog convertor (DAC) 32 and peripheral 34 (such as a monitor) from the DSP or from the drive controller.

FIG. 2 is a block diagram of an implementation of the drive controller 10 of the present invention. The key functional blocks are the DSP data and subcode interfaces 36 and 37, the buffer DRAM control 38, the error correction code (ECC) data corrector 40, the error detection and correction/cyclic redundancy checker (EDC/CRC) 42 and the host control 44. The DSP data interface descrambles and assembles data from the DSP 28, then stores the data into the RAM. The DSP subcode interface assembles subcode and stores P-W data into the RAM. A DSP address counter 58 generates an address for each block of data stored to the DRAM from the DSP interface. The error correction circuitry would first perform Reed-Solomon error correction on each block of data Reed-Solomon codes are random single- or multile-symbol error correcting codes operation on symbols which are elements of a finite field. All encoding, decoding, and correction computations are performed in the field. (See Practical Error Correction Design for Engineers, revised second edition, Cirrus Logic 1991 by Neal Glover and Trent Dudley). Then, a cyclic redundancy check of the corrected data would be performed. Since each codeword contains two parity bytes the drive controller of this invention can correct one error in each codeword. These ECC and EDC-CRC circuits are commonly available as hardware used in many other applications. The host control allows the corrected data to be transferred from the RAM to the host. Diagnostic data can be transferred from the host to the RAM, allowing testing of the ECC, EDC, host control RAM and system controller. Operation of the drive controller is controlled by a system controller through an 8-bit bus. The invention can decode CD media according to the Sony-Philips standard for CD-ROM and CD-I formats. These formats divide each 2 KB data block into two planes, each plane containing 43 P-codewords and 26 Q-codewords. Each codeword contains two parity bytes.

The host computer (not shown) is connected, through the IDE/ATA bus 46 and associated host interface, to the system controller 48 of the drive electronics of the CD drive and the host control 44. The host interface provides 8/16 bit peripheral input/output (PIO) and direct memory access (DMA) transfers of data to the host personal computer. The output buffers 54 of the invention can directly drive an IDE/ATA bus. The host interface also contains control and transfer status registers 56 accessible by the host. The design of the present invention allows the transfer of diagnostic data from the host to the RAM, allowing testing of the error correction circuitry, the host interface, the RAM itself and the system controller of the drive electronics.

The DRAM controller is connected to DRAM 50. The DRAM controller, under the direction of the host interface, accomplishes the transfer of data to the host and the error correction operations so as to insure an uninterrupted flow of data from the buffer RAM. The DRAM receives information from the DSP of the drive electronics via the DSP data and subcode logic interfaces. The DRAM also receives address information from the host microprocessor address counter 60, as well as receives corrected and addressed information from an ECC EDC address generator 52 which is connected with the error correction circuit ECC and the cyclic redundancy checker EDC CRC. Additionally, the DRAM stores header and subheader information to the header/subheader register 62.

Thus, the drive electronics system controller interface accepts digital data from the drive electronics in a serial stream, descrambles it, and assembles it into 8-bit bytes. It then stores the data into a DRAM buffer. The error correction and detection operations performed by the ECC and EDC CRC on each sector of data are managed by the DRAM controller, which insures, through the direction of the host interface, that a sector of data is being corrected while the transfer of previously corrected sectors of data is occurring in real-time and without interruption. The flow of data is controlled by a data path controller 64.

FIGS. 3a-3c are a pin diagram and accompanying pin-out assignments for an implementation of the present invention. The functions performed by this implementation will become clear by the following discussion.

FIG. 4 is a pin description of the system controller interface of an implementation of this invention. The system controller is the microcontroller that controls the operation of the IDE CD-ROM controller. The system controller interface contains an 8-bit bidirectional data transfer bus and is compatible with most microcontrollers.

FIG. 5a is a pin description of the host interface

of an implementation of this invention. This invention will support the ATAPI CD-ROM specification for an IDE CD-ROM interface. The drive controller can drive IDE interface signal lines directly. The host interface contains a 12 bytes command packet FIFO (first in first out) and IDE registers. These are used to direct or command the host interface by the host controller and to inform the host controller as to the precise status of the drive electronics. The host interface block also contains a data FIFO register for transferring data from the DRAM buffer to the host and vice versa. FIG. 5b is an address map of the host registers of an implementation of this invention. The logic conventions are as follows: A = signal asserted, N = signal negated.

FIG. 6 is a pin description of the DSP interface of an implementation of this invention.

This invention is designed to work with various DSP chips, which are selected using the DSPSL register. Serial data is received from the DSP. FIG. 7 is a pin description of the subcode interface of an implementation of this invention. P-W subcode information provided serially by the DSP will be stored into the data block of the DRAM simultaneously. This invention will support several different subcode interface protocols, which are selected using the SUBCD register. The command FIFO register COMIN is used to direct the host interface by the host controller. The status FIFO register SBOUT is used to inform the host controller as to the precise status of the drive electronics the HCON register informs the host interface as to which data format the host controller requires for the data to be place onto the IDE-ATA bus. The available data formats would include 8/16 bit peripheral input and output PIO and DMA. The TSTAT register allows the host controller to monitor the interface status. A reset timer allows the host controller to reset the system controller.

FIG. 8a is a pin description of the RAM interface of an implementation of this invention. This embodiment allows users to use conventional 128KB/256KB DRAM. FIG. 8b is a pin description for the RAM address bus of the RAM interface of an implementation of this invention. Note that several RAM address bits are reserved for future support of larger RAM sizes. FIG. 9 is a pin description of miscellaneous pins of an implementation of this invention.

Microcontroller Registers

The following map and description of the registers of an embodiment of this invention is intended for use in designs supporting the following configurations:

1. Type of RAM: 256K X 4 X 1 DRAM or 256K X 4 X 2 DRAM
2. Type of Host Interface: ATA (IDE) plus ATAPI CD-ROM

FIG. 10 is a description of the address register. The internal registers are indirectly addressed. The AR register holds the address of the register accessed with the subsequent R/W operation. The AR is read or written by the microcontroller if URS = 0. If URS = 1, the register addressed by the AR is read or written.

Except for address 00h (COMIN/SBOUT), the 4 least significant bits (bits 0 - 3) of the AR are automatically incremented following each read or write to any register For example, if the AR has been set to 2Ch, the AR automatically increments according to the following sequence during 18 consecutive reads or writes (with URS = 1): 2Ch, 2Dh, 2Eh, 2Fh, 20h, 21h, 22h, 23h, 24h, 25h, 26h, 27h, 28h. 29h, 2Ah, 2Bh, 2Ch, 2Dh.

Note: The AR does not automatically increment from 00h to 01h. Consecutive accesses to address 00h will repeatedly read (COMIN) or write (SBOUT). For example if the AR has been set to 0Ch, the AR automatically increments according to the following sequence during 7 consecutive reads or writes (with URS = 1): 0Ch, 0Dh, 0Eh,0Fh, 00h, 00h, 00h.

FIG. 11 is a description of command packet register. This register accesses the 12-byte Packet FIFO which receives commands or data from the host. The data transfer end interrupt (DTEIb) flag in the IFSTAT register is active (set to 0) while the 12-byte Packet FIFO is full. The command interrupt (CMDIb) flag in the IFSTAT register is active (set to 0) while one or more bytes from the host are present in the Packet FIFO. If an access from COMIN is attempted while the FIFO is empty, the value Ffh will be read.

Normally, flag DTEIb is used for receiving 12-byte packet commands from the host, and flag CMDIb is used for receiving data from the host (which may not be 12-bytes). Registers ATBHI and ATBLO (addresses 34h and 35h) can be used to control the number of consecutive bytes of DATA written into the Packet FIFO by the host. However, ATBHI and ATBLO should not be used to control the number of COMMAND bytes. Command or data writes from the host to the data port (1F0) are stored in the Packet FIFO if control bit Scod in register HICTL (20h.2) is set high. Note: An access to the COMIN register (00h) does not increment the AR.

UNUSED (01h-write) writes to address 00h in the controller accessed the SBOUT (status byte output) register. However, SBOUT is not useful for ATAPI operations. Address 00h should not be written to. Note: An access to address 00h does not increment the AR.

FIG. 12 is a description of interface status and interface control registers.

IFSTAT (Interface Status Register) holds the decoder and host interface status bits.

CMDIb - bit 7 - Command Interrupt flag is active-low with a logical 0 indicating an interrupt flag. "0" indicates that there are one or more bytes present in the

Packet FIFO and "1" indicates that the Packet FIFO is empty. Flag CMDIb is used for receiving data from the host, which may not be 12-bytes. For receiving 12-byte packet commands, flag DTEIb in register IFSTAT is used instead. CMDIb is automatically cleared (set to 1) after the last byte in the Packet FIFO is read. If control bit CMDIEN in the IFCTRL register (01h.7) is set high, pin UINTb (the microcontroller interrupt) will be active-low whenever the CMDIb flag is active-low.

DTEIb - bit 6 - Data-Transfer-End Interrupt flag is active-low with a logical 0 indicating an interrupt flag. "0" indicates that a host read from the Data FIFO or external RAM, or a 12-byte host write to the Packet FIFO (Packet FIFO full), is complete and "1" indicates that the interrupt flag has been cleared. DTEIb is automatically cleared when the microcontroller writes to the DTACK
register (07h). If control bit DTEIEN in the IFCTRL register (01h.6) is set
high, pin UINTb (the microcontroller interrupt) will be active-low whenever
the DTEIb flag is active-low.

DECIb - bit 5 - Decoder Interrupt flag is active-low with a logical 0 indicating an interrupt flag. "0" indicates that the decoder has finished processing a block and "1" indicates that the interrupt flag has been cleared. When the DECIb changes to active-low, the header registers (HEAD0-3), ECC block pointer registers (PTL, PTH), and status registers (STAT0-3) are ready to be read. If the ECC or EDC is enabled, DECIb changes to active-low at the completion of the EDC phase. If the ECC and EDC are disabled (write-only or disk-monitor operation), DECIb changes to active-low after the header registers (HEAD0-3) are ready. DECIb is automatically cleared (to 1) when the microcontroller reads the STAT3 register (0Fh).

X - bit 4 is undefined, and may return a 0 or 1.

DTBSYb - bit 3 - Data Transfer Busy flag is active-low with a logical 0 indicating a busy flag. "0" indicates that a data-transfer is in process and " 1 " indicates that no data-transfer in process. DTBSYb changes to active-low when the microcontroller writes to the data transfer trigger (DTTRG). DTBSYb is automatically cleared (to 1) when the host BEGINS to read the last byte to be transferred from the Data FIFO or external RAM.

X - bit 2 is undefined, and may return a 0 or 1. SBOUT is not useful for ATAPI operation.

DTENb - bit 1 - Date Enable is active-low with a logical 0. "0" indicates that a data-transfer is in process and "1" indicates that no data-transfer in process. After DTTRG is set, DTENb changes to active-low when the Data FIFO is ready to be read by the host. DTENb is automatically cleared (to 1) after the host reads the last byte to be uransferred from the Data FIFO or external RAM.

X - bit 0 is undefined. and may return a 0 or 1. SBOUT is not useful for ATAPI operation.

IFCTRL (Interface Control Register) provides control of the microcontroller interrupt and host interface.

CMDIEN - bit 7 - Command Interrupt Enable "1" allows pin UINTb (the microcontroller interrupt pin) to become active-low whenever the CMDIb flag in register IFSTAT is active-low. "0" inhibits the CMDIb flag from activating pin UINTb. CMDIEN controls the operation of pin UINTb. However, CMDIEN does not clear the interrupt request or control the CMDIb flag. CMDIEN is cleared to 0 by hardware reset or firmware reset.

DTEIEN - bit 6 - Data-Transfer-End Interrupt Enable "1" allows pin UINTb (the microcontroller interrupt pin) to become active-low whenever the DTEIb flag in register IFSTAT is active-low. "0" inhibits the DTEIb flag from activating pin UINTb. DTEIEN controls the operation of pin UINTb. However, DTEIEN does not clear the interrupt request or control the DTEIb flag. DTEIEN is cleared to 0 by hardware reset or firmware reset.

DECIEN - bit 5 - Decoder Interrupt Enable "1" allows pin UINTb (the microcontroller interrupt pin) to become active-low whenever the DECIb flag in register IFSTAT is active-low. "0" inhibits the DECIb flag from activating pin UINTb. DECIEN controls the operation of pin UINTb. However, DECIEN does not clear
the interrupt request or control the DECIb flag. DECIEN is cleared to 0 by hardware reset or firmware reset.

DOUTEN - bit 1 - Data Output Enable "1" enables host data reads from the Data FIFO or external RAM, or host writes to the Packet FIFO. "0" inhibits data transfers to the Data FIFO, external RAM, or Packet FIFO. Clearing DOUTEN (to 0) aborts data transfers to the FIFOs or external RAM. DOUTEN is cleared to 0 by hardware reset or firmware reset.

Unused Bits - bits 4, 3, 2, and 0 should only be set to 0.

FIG. 13 and FIG. 14 are descriptions of DBCL and DBCH (Data Byte/Word Counter) that form a 12-bit counter that controls or monitors the number of bytes or words transferred from the Data FIFO or external RAM to the Host. For 16-bit transfers, the number of WORDS minus one should be loaded into this counter. For 8-bit transfers, the number of BYTES minus one should be loaded. DBCH should always be written after DBCL is written, and zero should be written into bits 7-4 of DBCH. During the data transfer, the counter is decremented by one each time the host reads a word or byte. When reading DBCH, bits 7-4 each indicate the status of the data-transfer-end interrupt (DTEI), and have the same function (but opposite polarity) as the DTEIb flag in register IFSTAT. DBCL and DBCH are undefined following hardware reset or firmware reset.

FIG. 15, FIG. 16, FIG. 17 and FIG. 18 are descrip-

tions of HEAD0 to HEAD3 (Header Registers). Normally, these registers provide the header of each CD-ROM block, and are used to find the starting block during a disk seek. If control bit DECEN in register CTRL0 (0Ah.7) is enabled, the first four bytes (bytes 12-15) following each data sync are automatically stored in the header registers (HEAD0-3). During disk-monitor operation (see the description of register CTRL0), uncorrected header bytes are taken directly from incoming serial data If the incoming serial data is buffered, header bytes are taken from the buffer RAM, and are corrected if mode 1 is selected and ECC is enabled. In either case, HEAD0-3 should be read soon after the decoder interrupt occurs (bit DEC-Ib in register IFSTAT becomes 0). HEAD0-3 remains valid until the next sync occurs (see the description of register STAT3 for checking the valid time period). Generation of checkbytes during the authoring of CD-ROM disks includes ECC coverage of the header bytes for mode 1 blocks, but not for mode 2 blocks. Therefore if ECC is enabled, the header bytes are not valid unless the proper mode is selected using control bit MODRQ in register CTRL1 (0Bh.3). Operation of a mode 2 disk with mode 1 ECC causes the header bytes to be erased. By setting control bit SHDREN in register CTRL1 high, HEAD0-3 can be used to provide subheader bytes instead of header bytes. However, it is more convenient to use registers SUBH0-3 (14h-17h), which are not controlled by bit SHDREN, for this purpose. See the description of SUBH0-3 for subheader

information. Subheaders in HEAD0-3 follow the same format and operation as subheaders in SUBH0-3. HEAD0-3 are undefined following hardware reset or firmware reset.

FIG. 19 is a list of DACH, DACL settings for various starting points. DACL and DACH are Data Address Counters. DACL and DACH form a 16-bit counter that controls the buffer RAM address for transfers to the host. The microcontroller writes the starting address that corresponds to the required starting point in the CD-ROM block. After the starting address is set and register DTTRG is triggered, DACL and DACH are incremented automatically each time a byte or word is read by the host. The first byte of User Data is located at address 00h. DACH should always be written after DACL is written. For proper addressing, the mode of the CD-ROM block should be selected using control bit MODRQ in register CTRL1 (0Bh.3). DACL and DACH control the RAM address relative to the beginning of the block. The block number should also be specified, using data block registers DBACL and DBACH (24h and 25h). DACL and DACH are undefined following hardware reset or firmware reset.

DTTRG (Data Transfer Trigger) triggers the host transfer logic and prepares the Data FIFO, causing flag DTBSYb in register IFSTAT (01h.3) to become active-low. Before setting or triggering any data

transfer registers, control bit DOUTEN in register IFCTRL (01h.1) should be enabled. In the case of a host data read from the buffer RAM, triggering the transfer logic automatically fills the FIFO with data from the RAM. The count, RAM starting address, and block number should be set using registers DBCL, DBCH, DACL, DACH, DBACL, and DBACH (02h, 03h, 04h, 05h, 24h, and 25h) before triggering DTTRG. Flag DTENb in register IFSTAT (01h.1) becomes active-low when the FIFO first becomes ready. The microcontroller can also load registers UDTA0 - UDTA7, allowing host data reads (up to 8-bytes) from the microcontroller without using the buffer RAM. In this case the byte count, microcontroller data enable, and data bytes should be set using registers DBCL, DBCH, HDDIR, and UDTA0-7 (02h, 03h, 1Fh.6, and 40h-47h) before triggering DTTRG. After triggering DTTRG, trigger bit UDTRG in register HDDIR (1Fh.7) should be toggled to 1 followed by 0. For this type of transfer, flag DTENb in register IF-STAT (01h.1) has no meaning. Trigger DTTRG is not used for host writes to the Packet FIFO.

DTACK (Data Transfer Acknowledge) clears flag DTEIb to 1 in register IFSTAT (01h.6) and also clears the corresponding microcontroller interrupt (if enabled), terminating the data transfer sequence.

FIG. 19 and FIG. 20 are ECC block pointer/write address counters. PTL and PTH form a pointer used by the ECC logic, and contain the 12 least significant address bits of the first header byte of the CD-ROM block that is being corrected. Due to the DRAM page organization of one embodiment of the controller, the value of PTH,PTL will always be 00,00h, making it unnecessary to read or write PTL or PTH. The starting location of each block is controlled by write block counter registers WBKL and WBKH (28h and 29h). Error correction is processed on the block before that indicated in the write block counter (WBKH,WBKL - 1). The controller organizes the DRAM into 2048-byte pages, allowing PTL and PTH to remain unchanged. PTL and PTH are undefined following hardware reset or firmware reset.

WAL and WAH (Write Address Counter) form a 16-bit counter used by the write buffering logic. At the end of each data sync, WAH,WAL are automatically set to 00,00h. Following each word (two bytes) of write buffering into the external RAM, WAL and WAH are automatically incremented by two. Due to the DRAM page organization of the controller, WAL and WAH control the write location within each CD-ROM block, and are always set to 00,00h after each data sync. The starting location of each block is controlled by write block counter registers WBKL and WBKH (28h and 29h). It is not necessary to read or write WAL or WAH, except for debugging purposes. Because WAL and WAH are automatically incremented whenever control bits DECEN and WRRQ are enabled in register CTRL0 (0Ah.7 and 0Ah.2), WRRQ should be

disabled before reading the write address counter. WAH,WAL are cleared to 00,00h by hardware reset or firmware reset.

FIG. 22 is a description of WAL/CTRL0 (Control-0 Register). This register provides control of the ECC and write buffering logic.

DECEN - bit 7 - Decoder Enable "1" enables the decoding functions, allowing control bits E01RQ, AUTORQ, WRRQ, QRQ, and PRQ to control the ECC and write buffering logic. "0" disables the decoding functions, overriding control bits E01RQ, AUTORQ, WRRQ, QRQ, and PRQ. Changes to DECEN control the CD-ROM blocks following the next data sync. DECEN is cleared to 0 by hardware reset or firmware reset.

E01RQ - bit 5 - Error Detect and Correct Request "1" enables the error correction and detection (ECC and EDC) logic to process the following CD-ROM blocks, according to the settings of QRQ and PRQ. "0" disables the ECC and EDC logic. Changes to E01RQ control the CD-ROM blocks following the next data sync. If both QRQ and PRQ are enabled, the ECC/EDC sequence is Q-codewords, P-codewords, EDC-codeword. If QRQ is enabled but PRQ is disabled, the sequence is Q-codeword, EDC-codeword. If QRQ is disabled but PRQ is enabled, the sequence is P-codeword, EDC-codeword. If both QRQ and PRQ are disabled, only the EDC-codeword is checked. Normally, QRQ and PRQ are enabled whenever E01RQ is enabled in order to provide maximum correction capability. E01RQ is cleared to 0 by hardware reset or firmware reset.

AUTORQ - bit 4 - Automatic Correction Request "1" enables automatic error correction for mode 2 CD-ROM blocks, according to the setting of the FORM bit in the Subheader byte of each block. "0" disables automatic error correction for mode 2 CD-ROM blocks. In this case, error correction for mode 2 blocks is controlled by control bit FORMRQ in register CTRL1 (0Bh.2). Changes to AUTORQ control the CD-ROM blocks following the next data sync. AUTORQ does not control error correction in mode 1. AUTORQ is cleared to 0 by hardware reset or firmware reset.

WRRQ - bit 2 - Write Buffer Request "1" enables writes of incoming serial data to the external buffer RAM. automatically incremented when writes are enabled. "0" disables writes of incoming serial data to the external buffer DRAM.

If control bit SWEN is enabled in register CTRLW (10h.6), changes to WRRQ control writes following the next data sync. If SWEN is disabled, changes to WRRQ control writes immediately. Both WRRQ and SWEN are cleared to 0 by hardware reset or firmware reset.

QRQ - bit 1 - Q-codeword Correction Request "1" enables error correction of Q-codewords, allowing one error to be located and corrected within each Q-codeword. "0" disables error correction of Q-code-

words. Changes to QRQ control the CD-ROM blocks following the next data sync. QRQ is cleared to 0 by hardware reset or firmware reset.

PRQ - bit 0 - P-codeword Correction Request "1" enables error correction of P-codewords, allowing one error to be located and corrected within each P-codeword. "0" disables error correction of P-codewords. Changes to PRQ control the CD-ROM blocks following the next data sync. PRQ is cleared to 0 by hardware reset or firmware reset.

FIG. 23 is a description of CTRL0 DECODER OPERATION TABLE. NOTE: For repeated correction, see the description of register CRTRG (11h). For buffered-disk-monitor, see the description of control bit ROWEN in register CTRLW (10h.7).

FIG. 24 is a description of CTRL1 (Control-1 Register). This register provides control of the ECC and data sync logic.

SYIEN - bit 7 - Sync Insertion Enable "1" enables sync insertion, allowing the internal sync counter to provide timing if the block sync pattern in the incoming serial data contains errors. "0" disables sync insertion. By enabling both SYIEN and SYDEN, the internal sync counter can automatically provide timing if the sync pattern contains errors, and also re-synchronize whenever a new sync pattern is detected. SYIEN is cleared to 0 by hardware reset or firmware reset.

SYDEN - bit 6 - Sync Detection Enable "1" enables sync detection, allowing the internal sync counter to re-synchronize whenever a block sync pattern is detected in the incoming serial data "0" disables sync detection. SYDEN is cleared to 0 by hardware reset or firmware reset.

DSCREN - bit 5 - Descrambler Enable "1" enables the CD-ROM data descrambler. "0" disables the CD-ROM data descrambler. Disabling the descrambler is useful for reading uncompressed (Red Book) audio, or for debugging. Changes to DSCREN control the descrambler immediately. DSCREN is cleared to 0 by hardware reset or firmware reset.

COWREN - bit 4 - Correction Write Enable "1" enables bytes corrected by the ECC logic to be written to the external RAM. "0" disables bytes corrected by the ECC logic to be written to the RAM. By disabling COWREN, flags CRCOK and CBLK in registers STATO (0Ch.7) and STAT3 (0Fh.5) can be used to determine disk error rates. Changes to COWREN control the CD-ROM blocks following the next data sync. COWREN is cleared to 0 by hardware reset or firmware reset.

MODRQ - bit 3 - Mode Request "1" sets the error correction mode used by the ECC logic to mode 2. "0" sets the error correction mode used by the ECC logic to mode 1. After determining the mode from the incoming serial data control bit MODRQ must be set by the microcontroller. The raw mode data from the headers of the incoming serial data should' be read

from bits RMOD3-0 in register STAT2 (0Eh.7-4). If MODRQ is not set properly, the ECC logic will miscorrect. Note that operation of a mode 2 disk with mode 1 correction causes the header bytes to be erased. MODRQ is cleared to 0 by hardware reset or firmware reset. Generation of checkbytes during the authoring of CD-ROM disks includes ECC coverage of the header bytes for mode 1 blocks, but not for mode 2 blocks. Consequently, the mode byte in a mode 2 disk is not corrected. Mode changes are separated by pre-gap and post-gap blocks and track numbers. Changes to MODRQ control the CD-ROM blocks following the next data sync. MODRQ is cleared to 0 by hardware reset or firmware reset.

FORMRQ - bit 2 - Form Request "1" sets the form to 2, disabling the mode 2 ECC logic (but EDC is enabled). "0" sets the form to 1, enabling the mode 2 ECC logic. If control bit AUTORQ is enabled in register CTRL0 (0Ah.4), the setting of FORMRQ is not used by the ECC logic. FORMRQ is not used by the ECC logic if mode 1 is selected (control bit MODRQ set to 0 in register CTRL0). Changes to FORMRQ control the CD-ROM blocks following the next data sync. FORMRQ is cleared to 0 by hardware reset or firmware reset.

MBCKRQ - bit 1 - Mode Byte Check Request "1" enables checking of the mode byte. "0" disables checking of the mode byte. While checking of the mode byte is enabled, if the mode in the header of the incoming serial data does not match that selected by control bit MODRQ, ECC is disabled for the block and the NOCOR flag is set in register STAT0 (0Ch.5). Changes to MBCKRQ control the CD-ROM blocks following the next data sync. MBCKRQ is cleared to 0 by hardware reset or firmware reset.

SHDREN - bit 0 - Subheader Read Enable "1" selects subheader bytes to be provided by registers HEAD0-3 (04h-07h). "0" selects header bytes to be provided by registers HEAD0-3. By setting SHDREN high, HEAD0-3 can be used to provide subheader bytes instead of header bytes. However, it is more convenient to read the subheader from registers SUBH0-3 (14h-17h), which are not controlled by SHDREN. Changes to SHDREN control reads of HEAD0-3 immediately. SHDREN is cleared to 0 by hardware reset or firmware reset.

FIG. 25 is a description of STAT0 (Status-0 Register). This register provides status of the ECC, write buffering, and data sync logic.

CRCOK - bit 7 - Cyclic Redundancy Check OK. Flag CRCOK can become active-high only if the error detection (EDC) logic is enabled. This occurs automatically if correction or write-only decoder operations are selected (see the CTRL0 Operation Table) and "1" indicates that the cyclic redundancy check passed during the last ECC/EDC sequence. "0" indicates that the cyclic redundancy check failed during the last ECC/EDC sequence. Flag CRCOK becomes

valid when flag DECIb in register IFSTAT (01h.5) changes to active-low, and remains valid until the next block sync. See the description of flag VALSTb in register STAT3 (0Fh.7) to determine timing of the next sync. CRCOK is cleared to 0 by hardware reset or firmware reset.

ILSYNC - bit 6 - Illegal Sync flag can become active-high only if sync detection is enabled by control bit SYDEN in register CTRL1 (0Bh.6). In this case, occurrence of illegal sync re-synchronizes the internal sync counter. "1" indicates that a sync pattern was detected earlier than expected (less than 2352 bytes after the last detected or inserted sync). "0" indicates that no early sync pattern was detected. Flag ILSYNC becomes valid when flag DECIb in register IFSTAT (01h.5) changes to active-low, and remains valid until the next block sync. ILSYNC is cleared to 0 by hardware reset or firmware reset.

NOSYNC - bit 5 - No Sync flag can become active-high only if sync insertion is enabled by control bit SYIEN in register CTRL1 (0Bh.7). In this case, the internal sync counter will provide timing when the sync pattern is missing or has errors. "1" indicates that a sync pattern was not detected when expected (expected sync to occur 2352 bytes after the last detected or inserted sync). "0" indicates that a sync pattern was detected when expected. Flag NOSYNC becomes valid when flag DECIb in register IFSTAT (01h.5) changes to active-low, and remains valid until the next block sync. NOSYNC is cleared to 0 by hardware reset or firmware reset.

LBLK - bit 4 - Long Block flag can become active-high only if sync insertion is disabled by control bit SYIEN in register CTRL1 (0Bh.7). In this case, the internal sync counter will not provide timing when the sync pattern is missing or has errors. However, only 2352 bytes of incoming serial data will be written to the external RAM. "1" indicates that a sync pattern was not detected when expected (expected sync to occur 2352 bytes after the last detected sync). "0" indicates that a sync pattern was detected when expected. Flag LBLK becomes valid when flag DECIb in register IFSTAT (01h.5) changes to active-low, and remains valid until the next block sync. LBLK is cleared to 0 by hardware reset or firmware reset.

\WSHORT - bit 3 - Word Short "1" indicates that the incoming serial data rate exceeds the capability of the write buffering logic. "0" indicates that the incoming serial data rate was OK. The WSHORT error flag becomes valid immediately after the excessive rate is detected. This error is usually caused by hardware problems, and must be corrected for proper controller operation. WSHORT is cleared to 0 by hardware reset or firmware reset.

SBLK - bit 2 - Short Block flag can become active-high only if sync detection is disabled by control bit SYDEN in register CTRL1 (0Bh.6). In this case, occurrence of illegal sync will not re-synchronize the in-

ternal sync counter. "1" indicates that a sync pattern was detected earlier than expected (less than 2352 bytes after the last inserted sync). "0" indicates that no early sync pattern was detected. Flag SBLK becomes valid when flag DECIb in register IFSTAT (01h.5) changes to active-low, and remains valid until the next block sync. SBLK is cleared to 0 by hardware reset or firmware reset.

UCEBLK - bit 0 - Uncorrectable Errors in Block flag can become active-high only if the error correction (ECC) logic is enabled (Q-P, Q, or P-correction decoder operation selected). "1" indicates that one or more error bytes could not be corrected during the last ECC sequence. "0" indicates that no error bytes remained after the last ECC sequence. Flag UCEBLK becomes valid when flag DECIb in register IFSTAT (01h.5) changes to active-low, and remains valid until the next block sync. UCEBLK is cleared to 0 by hardware reset or firmware reset.

FIG. 26 is a description of STAT0 BLOCK SYNC STATUS TABLE.

FIG. 27 is a description of SSTAT1 (Status-1) register. This register provides erasure flags for the header and subheader bytes of the CD-ROM block. The erasure flags are provided through input pin C2PO.

HDERA - bit 4 - Header Erasure "1" indicates that the erasure flag was set for one or more header bytes. "0" indicates that no erasure flags were set for the header bytes.

SHDERA - bit 0 - Subheader Erasure "1" indicates that the erasure flag was set for both bytes in one or more subheader byte-pairs. "0" indicates that no erasure flags were set for both bytes in the subheader byte-pairs. During disk-monitor operation (see the description of register CTRL0), erasures are read directly from incoming C2PO flags. If the incoming serial data (from pin DSTATA) is buffered, the incoming C2PO flags are held and become available in STAT1 one block later, matching the one block delay of the buffered header and subheaders. In either case, HDERA and SHDERA become valid when flag DECIb in register IFSTAT (01h.5) changes to active-low, and remain valid until the next block sync.

FIG. 28 is a description of STAT2 (Status-2) register. This register provides mode and form information of the CD-ROM block.

FIG. 29 is a description of STAT2 RMODE TABLE. RMOD3-RMOD0 - bits 7-4 - Raw Mode provide mode information from the incoming serial data. during both buffer RAM and disk-monitor operation. Because RMOD3-RMOD0 cannot be changed by the ECC logic, they should be used for determining the mode of the CD-ROM block, according to the figure. N can be any number between 1 and 1Fh. RMOD3-RMOD0 become valid when flag DECIb in register IF-STAT (01h.5) changes to active-low, and remain valid until the next block sync. RMOD3-RMOD0 are

cleared to 0 by hardware reset or firmware reset.

MODE - bit 3 - Selected Mode flag provides the value of bit MODRQ in register CTRL1 (08h.3). "1" indicates that bit MODRQ is set high (mode-2 selected). "0" indicates that bit MODRQ is set low (mode-1 selected). Flag MODE becomes valid when flag DECIb in register IFSTAT (01h.5) changes to active-low, and remains valid until the next block sync. Flag MODE is cleared to 0 by hardware reset or firmware reset.

NOCOR - bit 2 - No Correction flag indicates whether error correction was performed NOCOR is valid only if control bit E01RQ, and QRQ or PRQ, are enabled in register CTRL0 (0Ah.5, 0Ah.1, 0Ah.0). "1" indicates that the last ECC/EDC sequence was aborted. "0" indicates that the last ECC/EDC sequence completed. The ECC/EDC sequence is aborted, and flag NOCOR set high, for the following reasons: Mode mismatch or erasure detected while control bit MBCKRQ is enabled in register CTRL1 (0Bh.1): A mode mismatch occurs if the mode in the header of the incoming serial data does not match that selected by control bit MODRQ in register CTRL1 (0Bh.3). A mode erasure occurs if the incoming C2PO flag is set for the fourth header byte, indicating unreliable mode data. Form 2 enabled while ECC logic is set to mode 2: Form 2 blocks cannot be corrected. Form 2 can be enabled by control bit FORMRQ in register CTRL1 (0Bh.2), or by the FORM bit in the Subheader byte if control bit AUTORQ is enabled in register CTRL0 (0Ah.4). FORM bit erasures while ECC logic is set to mode 2 and AUTORQ is enabled: A form bit erasure is detected if the incoming C2PO flags are set for both. FORM bits in the Subheader bytes. Illegal sync occurs while control bit SYDEN is enabled in register CTRL1 (0Bh.6), indicating that a sync pattern was detected earlier than expected. Control bit COWREN set low in register CTRL1 (0Bh.4). Flag NOCOR becomes valid when flag DECIb in register IFSTAT (01h.5) changes to active-low, and remains valid until the next block sync. Flag NOCOR is cleared to 0 by hardware reset or firmware reset.

RFORM1 - bit 1 - Raw Form Erasure "1" indicates that a form bit erasure was detected (a form bit erasure is detected if the incoming C2PO flags are set for both FORM bits in the Subheader bytes. "0" indicates that a form bit erasure was not detected. RFORM1 becomes valid when flag DECIb in register IFSTAT (01h.5) changes to active-low, and remains valid until the next block sync. RFORM1 is cleared to 0 by hardware reset or firmware reset.

RFORM0 - bit 0 - Raw Form Bit "1" indicates that the FORM bit was high in the Subheader bytes of the incoming serial data. "0" indicates that the FORM bit was low in the Subheader bytes of the incoming serial data. RFORM0 becomes valid when flag DECIb in register IFSTAT (01h.5) changes to active-low, and remains valid until the next block sync. RFORM0 is cleared to 0 by hardware reset or firmware reset.

UNUSED (0Eh-write) SBOUT is not useful for ATAPI. Address 0Eh should not be written to.

.FIG. 30 is a description of STAT3/RESET (Status-3) register. This register provides status of the ECC logic. Reading STAT3 clears flag DECIb to 1 in register IFSTAT (01h.6), and clears any active decoder interrupt.

VALSTb - bit 7 - Valid Status flag indicates the valid period during which the following header, pointer, and status registers can be read by the microcontroller: HEAD0-3 (04h-07h). PTL and PTH (08h-09h), STAT0-3 (0Ch-0Fh), SUBH0-3 (14h-17h), and WBKL and WBKH (28h and 29h). "1" indicates that the header, pointer, and status registers contain valid data, and are ready to be read. "0" indicates that the header, pointer, and status registers are not valid. Flag VALSTb becomes active-low when flag DECIb (decoder interrupt) in register IFSTAT (01h.5) changes to active-low, and returns high when the next block sync occurs (detected or inserted). Reading STAT3 does not change VALSTb. VALSTb is cleared to 1 by hardware reset or firmware reset.

CBLK - bit 5 - Corrected Block flag is valid only if the error correction (ECC) logic is enabled (Q-P,Q, or P-correction decoder operation selected). "1" indicates that one or more error bytes were corrected during the last ECC sequence. "0" indicates that no bytes were corrected during the last ECC sequence. Flag CBLK becomes valid when flag DECIb in register IFSTAT (01h.5) changes to active-low, and remains valid until the next block sync. CBLK is cleared to 0 by hardware reset, firmware reset, or by disabling WRRQ in register CTRL0 (0Ah.2).

RESET (Firmware Reset) activates firmware reset. Firmware reset clears most of the controller logic. However, to avoid disturbing important logic, firmware reset does not clear certain functions. The following list shows the differences between hardware reset (which clears all of the controller functions) and firmware reset. Functions NOT cleared by firmware reset: Clock stop logic controlled by input pin CLKSTP, register XTAL (1Ah) and output pin MCK, register DSPSL (1Bh), flag CS13 of register HDDIR (1Fh.6), register HICTL, register SUBC2 (21h), register RAMCF (2Ah), register MEMCF (2Bh), register SUBCD (2Ch), register UMISC (2Eh), register RSSTAT flags SRST, CMD, DIAGCMD, PARINT, RST, URST, and HRST (2Fh.7-5,3-0), R/W bit DRV in register ATDRS (36h) and in the ATAPI Drive Select Register, and control bits SRST and nIEN in the ATAPI Device Control Register. Flag URST of register RSSTAT (2Fh. 1) is set by firmware reset (see description of register RSSTAT).

FIG. 31 is a description of CTRLW (Control-Write) register. This register provides control of the write buffering logic. CTRLW bits 7, 3, 2, 1 and 0 should always be cleared to 0.

SWEN - bit 6 - Synchronized Write Enable "1" en-

ables synchronized write enable, causing changes by control bit WRRQ (0Ah.2) to be delayed until the end of the next block sync. "0" disables synchronized write enable. Selecting synchronized write enable causes the writing of incoming serial data to the buffer RAM to start or stop at the end of the next block sync. This prevents the writing of partial blocks into the RAM. SWEN should be changed only during decoder initialization. Write enable and disable is still controlled by bit WRRQ in register CTRL0 (0Ah.2). SWEN synchronizes changes in WRRQ to the end of sync, instead of randomly. SWEN is cleared to 0 by hardware reset or firmware reset.

SDSS - bit 5 - Subcode-DSP Sync Synchronization "1" enables subcode-DSP sync synchronization, causing audio write enables by control bit WRRQ (0Ah.2) to be delayed until the first left-channel lower-byte following the end of the subcode block. "0" disables subcode-DSP sync synchronization. Selecting Subcode-DSP Sync Synchronization causes the writing of incoming serial audio (red book) to the buffer RAM to start at the first left-channel lower-byte following the end of the subcode block. This prevents separate decoder and subcode interrupts from occurring. SDSS should be changed only during decoder initialization. Write enable and disable is still controlled by bit WRRQ in register CTRL0 (0Ah.2). SDSS synchronizes changes in WRRQ to the subcode block, instead of randomly. SDSS is cleared to 0 by hardware reset or firmware reset.

DCLKE - bit 4 - DSP Clock Enable "1" enables the incoming clock from the DSP. "0" disables the incoming clock from the DSP. DCLKE should be set high whenever DECEN in register CTRL0 is set high.

FIG. 32 is a description of CRTRG (Correction Retry Trigger) Writing 00h or 01h to register CRTRG triggers an error-correction retry.

CRTRL - bit 0 - Correction Retry Register Load "1" loads any updated E01RQ, QRQ, or PRQ values that have been written to register CTRL0 (0Ah.5,1,0) into the ECC sequencer, allowing the correction sequence to be changed. "0" does not load updated E01RQ, QRQ, or PRQ values into the ECC sequencer. Instead, the sequence of the last correction try is repeated.

Figures 33 through 38 ad SUBH0 to SUBH3 (Subheader Registers). These registers provide the subheader of each CD-ROM block, and operate similarly to header registers HEAD0-3. If control bit DECEN in register CTRL0 (0Ah.7) is enabled, data from the four pairs of bytes following each header (bytes 16-23) is automatically stored in the subheader registers (SUBH0-3). Bytes 16-23 are stored regardless of the setting of control bit MODRQ in register CTRL1 (0Bh.3), but the bytes contain subheader data only if mode 2 is selected (MODRQ = 1). During disk-monitor operation (see the description of register CTRL0), uncorrected subheader bytes are taken directly from in-

coming serial data. If the incoming serial data is buffered, subheader bytes are taken from the buffer RAM, and are corrected if form I (mode 2) is selected and ECC is enabled. In either case, SUBH0-3 should be read soon after the decoder interrupt occurs (bit DECIb in register IFSTAT becomes 0). SUBH0-3 remains valid until the next sync occurs (see the description of register STAT3 for checking the valid time period). The following figure shows the relationship between erasure flags and the byte numbers that are stored in SUBH0-3 (erasure flags are provided-!through input pin C2PO.

FIG. 39 is a description of Subheader Byte Number Table. SUBH0-3 areÿundefined following hardware reset or firmware reset.

FIG. 40 is a description of VER (Version) register. VER contains the version identification of the device. This register permits expansion and increased performance capabilities for future versions of the controller. VER is not changed by hardware reset or firmwase reset.

XTAL (Xtal) register provides control of the crystal frequeÿcy dividers.

MCK1 - bit 3 - Pin MCK 1x "1" sets the clock output at pin MCK to the crystal frequency by hardware reset, but is not changed by firmware reset.

XTALD2 - bit 0 - Crystal Divided by 2 "1"sets the internal controller clock to 1/2 crystal frequency. "0" sets the internal clock to the crystal frequency (no divider). XTALD2 is cleared to 0 by hardware reset, but is not changed by firmware reset.

FIG. 41 is a description of DSPSL (DSP Selection) register. This register selects the DSP configuration.

C2ML - bit 7 - C2 MSB to LSB "1" sets the direction of incoming erasures at pin C2PO to upper erasure followed by lower erasure. "0" sets the direction to lower erasure followed by upper erasure. C2ML is cleared to "0" by hardware reset, but is not changed by firmware reset.

SEL160 - bit 6 - Select 16 Offset "1" selects 16 bit-clocks per channel, with offset by one after LRCK. "0" does not select 16 bit-clocks with offset. SEL160 is cleared to "0" by hardware reset, but it is not changed by firmware reset.

LCHL - bit 5 - Left Channel Polarity "1" selects left channel as active if pin LRCK is "1". "0" selects left channel is active if pin LRCK is "0". LCHO is set to "1" by hardware reset, but is not changed by firmware reset.

SEL16 - bit 2 - Select 16 "1" selects 16 bit-clocks per channel. "0" does not select 16 bit-clocks. SEL16 is set to "1" by hardware reset, but is not hanged by firmware reset.

DIR - bit 1 - Data Direction "1" selects the rising edge of DBCK for latching incoming data at pin DSDATA. "0" selects the falling edge of DBCK for latching DSDATA. If the incoming data at pin DSDATA

changes at the falling edge of DDBCK, use the rising edge for latching' if DSDATA changes at the rising edge of DBCK, use the falling edge for latching. EDGE is set to 1 by hardware reset, but is not chanted by firmware reset.

Fig. 42 DSP SELECTION TABLE shows the settings for various DSPs. the default setting after hardware reset is 00100101 (Matsushita MN66261). The setting of DSPSL is not changed by firmware reset.

FIG. 43 is a description of the HCON/UACL register.

FIG. 44 is a description of the DSPSL register.

FIGS. 45, 49 and 50 are descriptions of the UACL, UACH and UACU Microcontroller- RAM Address Counter which forms a 20-bit counter that controls the buffer address for transfers between the microcontroller and RAM. The counter can be set to any physical location in the. buffer RAM, and contains enough bits to support larger RAM sizes in future revisions. After waiting for busy flag URTBSY to be low in register HDDIR (1Fh.7), the microcontroller writes the RAM starting address into the counter. UACL, UACH, and UACU are incremented automatically each time a byte is read or written. See the description of registers RAMRD, (1Eh), RAMWR (1Eh), and flag URTBSY in register HDDIR (1Fh.7). UACH should always be written after UACL is written, and UACU should always be written after UACH is written. UACL, UACH and UACU are undefined following hardware reset or firmware reset.

FIG. 51 is a description of the RAMRD/RAMWR RAM Read and Write registers. The microcontroller accesses the buffer RAM by reading from the RAMRD register or writing to the RAMWR register. To initialize a read or write sequence, the microcontroller waits for busy flag URTBSY to be low in register HDDIR (1Fh.7), then writes the RAM starting address into the counter formed by UACL (1Ch), UACH (1Dh), and UACU (2Dh). Reading RAMRD causes events (1), (2) and (3) to occur in the following order: (1) - data previously stored in RAMRD is transferred to the microcontroller: (2) - RAM data at the counter address is transferred to the RAMRD register, and (3) counter UACL, UACH, and UACU is incremented and flag URTBSY cleared. After the RAM starting address is written to the counter, the first read of register RAMRD will transfer an INVALID byte to the microcontroller, followed by the starting byte from the RAM to the RAMRD register. The invalid byte remains from a previous access, or from power-up. Because the counter is automatically incremented, sequential reads can be used without writing new addresses into UACL, UACH, and UACU. However, flag URTBSY should be checked before each sequential read from RAMRD to make sure that events (1), (2) and (3) in the previous transfer from RAM to RAMRD have completed.

Writing RAMWR causes the following events to

occur in the following order: (1) - data is transferred from the microcontroller to register RAMWR; (2) - data is transferred from RAMWR to the RAM (at the counter address); and (3) - counter UACL, UACH and UACU is incremented and flag URTBSY is cleared. After the RAM starting address is written to the counter, the first write to register RAMWR will transfer a VALID byte to the RAM. Because the counter is automatically incremented, sequential writes can be used without writing hew addresses into UACL, UACH and UACU. However, flag URTBSY should be checked before each sequential write to RAMWR to make sure that events (1) - (3) in the previous transfer from RAMWR to RAM have completed. The contents or RAMRD and RAMWR are undefined following hardware reset or firmware reset.

FIG. 52 is a description of HDDIR Host data Direction Register. This register provides microcontroller and host transfer flags and control.

URTBSY - read bit 7 - Microcontroller to RAM Transfer Busy "1" indicates that the previous microcontroller-RAM transfer is in progress. "0" indicates that the microcontroller-RAM transfer logic is not busy. URTBSY is cleared to 0 by hardware reset or firmware reset.

CS13 - read bit 6 Chip Select 1 and 3 "1" indicates that input pins CS1FX- and CS3FX- became active at the same time, indicating present of a non-ATA host adaptor. "0" indicates normal operation. Flag CS13 can be used to support non-ATA host adapters that have lines CS1FX- and CS3FX- connected together. With this adaptor configuration, host writes to the ATAPI Features Register (1F1) will set flag CS13, allowing firmware to respond appropriately to adapters that do not support separate CS3FX-addressing. CS13 is cleared to 0 by hardware reset, but is not changed by firmware reset.

UDTRG and UDATA - write bits 7 and 6 - Microcontroller Data Trigger/Select are normally set to "o", selecting data transfers from the buffer RAM to the host. Setting UDATA to "1" enables microcontroller writes to data registers UDTA0-7 (40h-47h), and allows high-speed 8-bit or 16-bit data transfers from UDTA0-7 to the host. Writing to UDTRG triggers the transfer from UDTA0-7 to the host. This type of transfer is efficient for the small amounts of data (up to eight bytes can be transferred at a time). Registers IFSTAT, IFCTRL, DBCL, DBCH, DTTRG, and DTACK (01h-03h, 06h and 07h) are used in the same way as a RAM to host transfer, except flag DTENb in IFSTAT has no meaning. However, registers DACL and DACH are not used. After enabling control bit DOUTEN (in register IFCTRL), loading DBCL, DBCH and setting UDATA to 1, and writing to registers UDTA0-7, the microcontroller writes to register DTTRG. Next, the microcontroller sets UDTRG to 1, followed by 0, to trigger the FIFO-ready transfer logic. The host will receive data beginning with UDTA0 and ending with

UDTA7. UDTRG and UDATA are cleared to 0 by hardware reset or firmware reset.

HOST16 - write bit 5 - Host 16-bit Select "1" selects 16-bit Data reads and Packet FIFO writes (at host register 1F0h). "0" selects 8-bit Data reads and packet-FIFO writes. because 8-bit data transfers do not conform to the ATAPI specification, HOST16 should normally be set to 1. HOST16 is cleared to 0 by hardware reset or firmware reset. Note: For 16-bit data reads, the number of WORDS minus one should be loaded into DBCL and DBCH. For 8-bit data reads, the number of BYTES minus one should be loaded into DBCL and DBCH.

FIG.53 is a list of the only values that should be written to HDDIR - write bits 4, 3, 2, 1 and 0. following hardware or firmware reset.

FIG.54 is a description of the HICTL Host Interface Control register. This register provides control of the host interface.

HICTL - bit 7 should only be set to "0" This bit is cleared to 0 by hardware reset but is not changed by firmware reset.

PDIAGEN - bit 6 - Pin HPDIAG- Enable "1" sets pin HPDIAG- to the active-low state. "0" clears HPDIAG- to the high-impedance state (HPDIAG- is an open-drain pin). PDIAGEN is automatically cleared to 0. clearing pin HPDIAG- to high-impedance by hardware reset command Execute Drive Diagnostics (ATA opcode 90h), or ATA Soft Reset (SRST). After PDIAGEN is automatically cleared, pin HPDIAG- should be set following the timing in ATAPI and ATA specification. Execute Drive Diagnostics and ATA Soft Reset clear PDIAGEN even if the drive is not selected in the ATAPI Drive Select Register. PDIAGEN is not changed by firmware reset.

DASPEN - bit 5 - Pin HDASP- Enable "1" sets pin HDASP- to the active-low state. "0" clears HDASP- to the high-impedance state (HDASP- is an open drain pin). DASPEN is automatically cleared to 0, clearing pin HDASP- to high-impedance by hardware reset command Execute Drive Diagnostics (ATA opcode 90h), or ATA Soft Reset (SRST). After DASPEN is automatically cleared. pin HDASP- should be set following the timing in ATAPI and ATA specification. Execute Drive Diagnostics and ATA Soft Reset clear DASPEN even if the drive is not selected in the ATAPI Drive Select Register. DASPEN is not changed by firmware reset.

CLRBSY - bit 4 - Clear BSY "1" prepares the clearing logic for flag BSY in the ATAPI Status Register. Note: BSY is actually cleared by the FOLLOWING write to register HICTL. "0" should be written to CLRBSY during the FOLLOWING write. Whenever flag BSY in the ATAPI Status Register (1F7h) is set, whether automatically or by control bit SETBSY, BSY should be cleared (using CLRBSY) as soon as allowed by the ATAPI and ATA specifications. See the description of control bit SETBSY. CLRBSY is not

changed by firmware reset.

SETBSY - bit 3 - Set BSY "1" prepares the setting logic for flag BSY in the ATAPI Status Register. Note: BSY is actually set by the FOLLOWING write to register HICTL. "0" should be written to SETBSY during the FOLLOWING write. Writing 1 to SETBSY activates the microcontroller interrupt, if enabled by control bit IDEIEN in register UMISC (2Eh.7). Flag BSY is automatically set, and the microcontroller interrupt activated, by hardware reset, command Execute Drive Diagnostics (ATA opcode 90h), ATA Soft Reset (SRST), or any command written to the ATAPI Command Register (host register 1F7h). SETBSY and flag BSY are not changed by firmware reset.

SCOD - bit 2 - Select command Packet or Data "1" selects the Command-Packet FIFO to be addressed by the ATA data port (host address 1F0h). "0" selects the buffer RAM to be addressed by the ATA data port. SCOD is cleared to 0 by hardware reset, but is not changed by firmware reset. IORDYEN - bit 1 - Pin IORDY Enable "1" allows the data transfer logic to de-assert pin IORDY whenever necessary. "0" does not allow IORDY to be de-asserted)IORDY acts . as an open-drain pin). If IORDYEN is set high. pin IORDY will be de-asserted to 0 whenever it is necessary to slow down the data transfer rate to match the capability of the CD-ROM drive. The capability of the drive depends on the RAM configuration and crystal frequency. Note" Some systems will not work properly if pin IORDY is de-asserted IORDYEN is set to 1 by hardware reset but is not changed by firmware reset.

IOCS16EN - bit 0 Pin IOCS16- Enable "1" allows pin IOCS16- to become active low during 16-bit reads from the buffer RAM or 16-bit writes to the Packet FIFO (at host register 1F0h). Note: Control bit HOST16 in register HDDIR (1Fh.5) must also be enabled. "0" does not allow IOCS16- to be asserted (IOCS16- is an open drain pin). Because both IOCS16EN and HOST16 must be enabled to allow assertion of pin IOCS16-, IOCS16EN can always be set to 1. In this case. HOST16 will correctly control pin IOCS16- and all 16-bit transfer logic. To conform to the ATAPI specification, 16-bit data transfers should be used. IOCS16EN is cleared to 0 by hardware reset. but is not changed by firmware reset.

FIG. 55 is a description of SUBC2 Subcode Control-2 register. This register provides control of the subcode interface.

NOPQ - bit 3 No P-data or Q-data, "1" clears (to 0) bits 7 and 6 (P-data and Q-data) of subcode data that is written to the buffer RM. "0" allows the P-data and Q-data bits to be included in subcode data that is written to the buffer RAM. NOPQ is cleared to 0 by hardware reset, but is not changed by firmware reset.

FIG. 56 is a description of the DSP Subcode Clock TABLE.

CDSP2, CDSP1, and CDSP0 - bits 2, 1 and 0 - DSP Subcode Clock Select, if subcode is buffered the DSP clock select bits must be set as shown in the figure, in order to match the subcode data rate. Only the combinations shown in the table should be used. DSP2-0 only control subcode clocking logic, and do not need to be set unless subcode is written to the buffer RAM. CDSP2-0 are cleared to 0 by hardware reset, but are not changed by firmware reset.

FIG. 57 is the STATS Status of subcode register. If read from, STATS provides status of the subcode interface. If written to, STATS clears the subcode interrupt (if enabled) and status flags.

STATS bits 7, 6, 5, 4 and 3 are undefined During reads of STATS by the microcontroller, bits 7-3 are undefined, and can be high or low.

MISSY - bit 2 - Missing Subcode Sync "1" indicates a missing subcode sync condition. "0" indicates subcode sync is not missing. Flag SINT in register RSSTAT (2Fh.4) is set (to 1) whenever flag MISSY is set. If enabled by control bit SCIEN in register SUBCD (2Ch.4), a microcontroller interrupt is also activated when MISSY is set. SINT, interrupt, and MISSY are cleared by writing any value to STATS.

SBKEND - bit 1 - Normal End of Subcode Block "1" indicates a normal subcode block end. "0" indicates no normal subcode block end. Flag SINT in register RSSTAT (2Fh.4) is set (to 1) whenever flag SBKEND is set. If enabled by control bit SCIEN in register SUBCD (2Ch.4), a microcontroller interrupt is also activated when SBKEND is set. SINT, interrupt, and SBKEND are cleared by writing any value to STATS.

SILSY - bit 0 - Illegal Subcode Sync "1" indicates . a normal subcode block end. "0" indicates no normal subcode block end. Flag SINT in register RSSTAT (2Fh.4) is set (to 1) whenever flag SILSY is set. If enabled by control bit SCIEN in register SUBCD (2Ch.4), a microcontroller interrupt is also activated when SILSY is set. SINT, interrupt, and SILSY are cleared by writing any value to STATS.

FIGS. 59 and 60 are descriptions of DBACL and DBACH Data Transfer Block Registers. In order to free the microcontroller from calculating 2352-byte address boundaries, the buffer RAM is partitioned into blocks. Registers DBACL and DBACH control the RAM block number of the data to be transferred, while counters DACL and DACH (04h and 05h) control the address relative to the beginning of the RAM block specified by DBACL and DBACH. The RAM block number is not incremented automatically, and must be set before each block transfer to the host begins. DBACL and DBACH contain enough bits to support larger RAM sizes in future revisions. For data transfer information, see the description of registers IFCTRL, DBCL, DBCH, a=DACL, DACH DTTRG, and DTACK (01h-07h). DBACH should always be written after DBACL is written. DBACL, DBACH are cleared to 00,00h by hardware reset or firmware reset.

FIGS. 61 and 62 are descriptions of SBKL and

SBKH Subcode Write Block Registers. After the appropriate interrupt occurs, registers SBKL and SBKH point to the RAM block number of subcode that is available for transfer to the host. Also, the number in SBKL and SBKH plus 1 points to the RAM block number of the buffer area for writing incoming subcode. Register SBADR (23h) controls the write location within each block. The RAM block number in SBKL and SBKH is incremented automatically, and only needs to be updated by the microcontroller in order to overwrite a discarded block. SBKL and SBKH contain enough bits to support larger RAM sizes in future revisions. If subcode sync and data sync are synchronized by enabling control bit SDSS in register CTRLW (10h.5), SBKL and SBKH should be read soon after the decoder interrupt occurs (bit DECIb in register IFSTAT becomes 0). In this case, SBKL and SBKH remain valid until the next data sync occurs (see the description of register STAT3 for checking the valid time period). If subcode sync and data sync are not synchronized, SBKL and SBKH should be read soon after the subcode interrupt occurs (bit MISSY, SBKEND, or SILSY in register STATS becomes 1). In this case, SBKL and SBKH remain valid until the next subcode interrupt occurs. The value read from the lower 9 bits of SBKL and SBKH during the valid time period identifies the RAM block that is available for transfer to the host. During reads of SBKH by the microcontroller, bits 7-1 are undefined, and can be high or low. SBKH, SBKL are cleared to 00,00h by hardware reset or firmware reset, causing buffering of incoming subcode data to begin at block number 1.

FIGS. 63 and 64 are descriptions of WBKL and WBKH Decoder and Buffer-Write Block Counter registers. In order to free the microcontroller from calculating 2352-byte address boundaries, the buffer RAM is partitioned into blocks. Registers WBKL and WBKH point to the RAM block number of the data to be processed by the error correction logic. Also, the number in WBKL and WBKH plus 1 points to the RAM block number of the buffer area for writing incoming serial data Registers WAL and WAH (08h/0Ah and 09h/0Bh) control the write location within each RAM block. The RAM block number in WBKL and WBKH is incremented automatically, and only needs to be updated by the microcontroller in order to overwrite a discarded block. WBKL and WBKH contain enough bits to support larger RAM sizes in future revisions. WBKL and WBKH should be read soon after the decoder interrupt occurs (bit DECIb in register IFSTAT becomes 0). WBKL and WBKH remain valid until the next sync occurs (see the description of register STAT3 for checking the valid time period). The value read from the lower 9 bits of WBKL and WBKH during this valid time period identifies the RAM block that is available for transfer to the host. During reads of WBKH by the microcontroller, bits 7-1 are undefined, and can be high or low. WBKH, WBKL are cleared to 00,00h by hard-

ware reset or firmware reset, causing buffering of incoming serial data to begin at block number 1.

FIG. 65 is a description of RAMCF RAM Configuration Register. This register provides control of the RAM interface configuration. Each control bit written to this register can also be read by the microcontroller.

RFTYP - bit 7 Refresh Type "1" selects CAL before RAM DRAM refresh. "0" selects CAS only DRAM refresh. RFTYP is cleared to 0 by hardware reset, but is not changed by firmware reset.

RAMCLR - bit 6 - RAM Clear Enable "1" enables RAM clearing, filling all locations in the buffer RAM. "0" disables RAM clearing. To clear the RAM, first write the clear data value (normally 00h) to register RAMWR (1Eh). Next, write 00h to registers UACL, UACU, and UACH (1Ch, 1Dh and 2Dh). Enable RAMCLR to begin writing the contents of RAMWR to each RAM location. When all RAM locations have been filled, RAM0 (bit 5) will change from 0 to 1. After RAM clearing has completed, the microcontroller should clear RAMCLR to 0. RAM0 will return to 0 when RAMCLR is disabled. If enabled by control bit RPEN, the correct parity bit will be written to all locations during RAM clearing. RAMCLR is cleared to 0 by hardware reset, but is not changed by firmware reset.

RAM0 - bit 5 - RAM Clear Flag (read only) "1" indicates that the RAM address has wrapped around beyond 00,00,00h, and that RAM clearing has completed. "0" indicates that the RAM clearing has not completed or is disabled. RAM0 is cleared to 0 by hardware reset, but is not changed by firmware reset.

UHILO - bit 4 - Host High-Low Swap "1" enables byte swaps for reads from the buffer RAM to the host, causing odd bytes to be read before even bytes. "0" disables byte swap, causing evenbytes to be read before odd bytes. Byte swapping is not normally needed for ATAPI operation. UHILO is cleared to 0 by hardware reset, but is not changed by firmware reset.

RPEN - bit 3 RAM Parity Enable "1" enables parity checking, and parity interrupt, of DRAM data. "0" disables parity checking and parity interrupt. Enabling RAM parity allows defective DRAMS to be detected. RAM clearing, using control bit RAMCLR, should be completed before RAM parity is enabled. If a parity error is detected, flag PARINT will be set to 1 in register RSSTAT (2Fh.3) and the microcontroller interrupt pin will be activated. Flag PARINT and the interrupt are cleared by clearing RPEN to 0. RPEN is cleared to 0 be hardware reset, but is not changed by firmware reset.

FIG. 66 is a table of RCF2, RCF1 and RCF0 - bits 2, 1 and 0 - RAM Configuration. Only the settings shown in the RAM Configuration Table should be used. RCF2, RCF1, and RCF0 are cleared to 0 by hardware reset, but are not changed by firmware reset.

FIG. 67 is a description of MEMCF (Memory Layout Configuration) register. This register provides control of the RAM layout configuration.

MEMCF bits 7, 6, 5, and 4 must always be set to 0. MEMCF bits 7, 6, 5, and 4 are cleared to 0 by hardware reset, but are not changed by firmware reset.

PURG - bit 3 - Data FIFO Purge "1" purges the RAM to Host Data FIFO. "0" disables the FIFO purge logic. This bit should be cleared to 0 except during the purge operation. PURG is cleared to 0 by hardware reset, but is not changed by firmware reset.

IORDYF - bit 2 - Pin IORDY Fast Enable "1" enables output pin IORDY to be de-asserted to 0 without qualification by input pin HRD-. "0" disables unqualified de-assertion of pin IORDY. This bit speeds up de-assertion of pin IORDY by ignoring the state of pin HRD-. However, unqualified de-assertion of IORDY violates the ATA specification, and may interfere with normal operation of many systems. IORDYF is cleared to 0 by hardware reset, but is not changed by firmware reset.

FIG. 68 is a description of MLY1 and MLY0 - bits 1 and 0 - Memory Layout Configuration Following hardware reset, the memory layout configuration should be set as shown in the figure. Only the settings shown in the Memory Layout Configuration Table should be used. Normally, all auxiliary blocks are buffered (MLY1 = 1 and MLY0 = 1). If only the last 2 auxiliary blocks are buffered, the configuration must be changed whenever data transfer sizes above 2048 bytes are required. MLY1 and MLY0 ARE CLEARED TO 0 BY HARDWARE RESET, but are not changed by firmware reset.

FIG. 69 is a description of SUBCD Subcode Control register. This register provides control of the subcode interface.

SBXCK - bit 7 - Subcode External Clock "1" selects an external clock from pin EXCK to be used by the subcode logic. "0" selects an internal clock to be used by the subcode logic. SBXCK is cleared to 0 by hardware reset, but is not changed by firmware reset.

SCEN - bit 6 - Subcode Enable "1" enables the subcode logic. "0" disables the subcode logic. SCEN is cleared to 0 by hardware reset, but is not changed by firmware reset.

SCKB2 - bit 5 - Subcode Clock Divide By 2 "1" enables the divide by 2 logic (for internal or external clock). "0" disables the divide by 2 logic. SCKB2 is cleared to 0 by hardware reset, but is not changed by firmware reset.

SCIEN - bit 4 - Subcode Interrupt Enable "1" enables activation of subcode interrupts to the microprocessor. "0" disables subcode interrupts. SCIEN is cleared to 0 by hardware reset. but is not changed by firmware reset.

EXINV - bit 3 - External Clock Invert Select "1" selects an inverted output clock at pin EXCK if EXCK is set as an output. "0" selects a non-inverted clock. EX-

INV is cleared to 0 by hardware reset, but is not changed by firmware reset.

EXOP - bit 2 - Pin EXCK Operation "1" sets EXCK as an output pin. "0" sets EXCK as an input pin. EXOP is cleared to 0 by hardware reset, but is not changed by firmware reset.

FIG. 70 is SBSEL1 and SBSEL0 - bits 1 and 0 - Subcode Format Select Table. The subcode format should be set according to the figure. SBSEL1 and SBSEL0 are cleared to 0 by hardware reset, but are not changed by firmware reset.

FIG. 71 is a description of UMISC (Miscellaneous Microcontroller Control) register.This register provides miscellaneous flags and control bits.

PDIAGb - read-bit 1 - Pin HPDIAG- Flag "1" indicates that open-drain pin HPDIAG- is high (inactive). In this case. both master and slave drives are de-asserting pin HPDIAG-. "0" indicates that pin HPDIAG- is low (active) In this case, either the master or slave drive is setting pin HPDIAG- to active-low.

DASPb - read-bit 0 - Pin HDASP- Flag "1" indicates that open-drain pin HDASP- is high (inactive). In this case, both master and slave drives are de-asserting pin HDASP-. "0" indicates that pin HDASP- is low (active). In this case, either the master or slave drive is setting pin HDASP- to active-low.

IDEIEN - write-bit 7 - IDE Interrupt Enable "1" enables activation of IDE (ATA) interrupts to the microprocessor. "0" disables activation of IDE (ATA) interrupts. IDE interrupts (if enabled) are activated, and flag SRST, CMD, DIAGCMD, or HRST set in register RSSTAT (2Fh.7,6,5,0), whenever the BSY flag in the ATAPI status register is set automatically by: 1 written to bit SRST (Soft Reset) in the ATAPI Device Control Register (host address 3F6h) in the master or slave drive. Note: The BSY flag and IDE interrupt (if enabled) cannot be cleared while SRST is set to 1. Any command written to the ATAPI Command Register (host address 1F7h) while the drive is selected. Command Execute Drive Diagnostics (ATA opcode 90h) written to the master or slave drive. Note: if opcode 90h is written while the drive is selected, both flags CMD and DIAGCMD will be set.

Hardware Reset (however, hardware reset clears IDEIEN). Writing 1 followed by 0 to SETBSY in register HICTL (20h.3) sets BSY and activates the interrupt (if enabled by IDEIEN) but does not set a status flag. The BSY flag and IDE interrupt are cleared by writing 1 followed by 0 to control bit CLRBSY in register HICTL(20h.4).IDEIEN is cleared to 0 by hardware reset, but is not changed by firmware reset.

UMISC write-bit 6 should always be cleared to 0.

DRVEb - write-bit 5 - Drive Enable "1" disables selection of the drive, whether bit DRV in the ATAPI Drive Select Register (host address 1F6h) is 0 or 1. "0" enables selection of the drive if bit DRV matches the setting of DRVIb in register UMISC. DRVEb is set to 0 (active) by hardware reset, but is not changed by

firmware reset.

DRV1b - write-bit 4 - Drive 1 "1" sets the drive to be selected when bit DRV in the ATAPI Drive Select Register is set to 0 (drive 0). "0" sets the drive to be selected when bit DRV in the ATAPI Drive Select Register is set to 1 (drive 1). DRV1b is set to 0 (drive 1) by hardware reset, but is not changed by firmware reset.

HINTRQ - write-bit 3 - Host Interrupt Request "1" sets pin HIRQ high if the drive is selected and NIEN (Interrupt Enable) is enabled in the ATAPI Device Control Register (host address 3F6h). "0" clears pin HIRQ (to 0) if the drive is selected and NIEN is enabled. HINTRQ is automatically cleared to 0 by the following: Hardware reset, 1 written to bit SRST (Soft Reset) in the ATAPI Device Control Register (host address 3F6h) in the master or slave drive, any command written to the ATAPI Command Register (host address 1F7h) while the drive is selected, and a read from the ATAPI Status Register (host address 1F7h) while the drive is selected. HINTRQ is not changed by firmware reset, or by reads from the ATAPI Alternate Status Register (host address 3F6h). If the drive is not selected, or if NIEN is disabled (cleared to 1), pin HIRQ becomes high-impedance.

UMISC write-bits 2, 1, and 0 should always be set to 0.

RSSTAT (Reset, IDE, and Subcode Status) register provides status flags for reset, IDE, and subcode logic.

SRSTF - bit 7 - Soft Reset Flag "1" indicates that 1 has been written to bit SRST (Soft Reset) in the ATAPI Device Control Register (host address 3F6h) in the master or slave drive. "0" indicates that 1 has not been written to bit SRST. The BSY flag is set, and IDE interrupt to the microcontroller activated (if enabled), whenever SRST is set. BSY and IDE interrupt cannot be cleared until SRST is cleared to 0 (however, the IDE interrupt can be disabled). After SRST is cleared, the BSY flag and IDE interrupt are cleared by writing 1 followed by 0 to control bit CLRBSY in register HICTL(20h.4). Flag SRSTF is cleared to 0 by hardware reset, but is not changed by firmware reset.

CMD - bit 6 - ATA Command "1" indicates that a command has been written to the ATAPI Command Register (host address 1F7h) while the drive was selected. "0" indicates that a command has not been written. The BSY flag is set, and IDE interrupt to the microcontroller activated (if enabled), whenever a command is written to the ATAPI Command Register while the drive is selected. The BSY flag and IDE interrupt are cleared by writing 1 followed by 0 to control bit CLRBSY in register HICTL (20h.4). CMD is cleared to 0 by hardware reset, but is not changed by firmware reset.

DIAGCMD - bit 5 - Execute Drive Diagnostics Command "1" indicates that the ATA command Execute Drive Diagnostics (ATA opcode 90h) has been written to the master or slave drive. "0" indicates that Execute Drive Diagnostics has not been written. The BSY flag is set, and IDE interrupt to the microcontroller activated (if enabled), whenever Execute Drive Diagnostics is written to the ATAPI Command Register. If opcode 90h is written while the drive is selected, both flags CMD and DIAGCMD will be set. The BSY flag and IDE interrupt are cleared by writing 1 followed by 0 to control bit CLRBSY in register HICTL (20h.4). DIAGCMD is cleared to 0 by hardware reset, but is not changed by firmware reset.

SINT - bit 4 - Subcode Interrupt Flag "1" indicates that flag MISSY, SBKEND, or SILSY has been set in register STATS (22h.2,1,0). "0" indicates that flag MISSY, SBKEND, or SILSY has not been set. If enabled by control bit SCIEN in register SUBCD (2Ch.4), a microcontroller interrupt is activated when SINT is set. SINT, interrupt, and the flag in register STATS are cleared by writing any value to STATS. SINT is cleared to 0 by hardware reset or firmware reset.

PARINT - bit 3 - Parity Interrupt Flag "1" indicates that a parity error has been detected in the DRAM. "0" indicates that a parity error has not been detected. If enabled by control bit RPEN in register RAMCF (2Ah.3), a microcontroller interrupt is activated when PARINT is set. PARINT and the interrupt are cleared by writing 0 to RPEN. PARINT is cleared to 0 by hardware reset, but is not changed by firmware reset.

RST - bit 2 - Reset Flag "1" indicates that the device is currently being reset. "0" indicates that the device is not currently being reset. RST allow the hardware reset to be monitored (if the microcontroller is not reset at the same time).

URST - bit 1 - Firmware Reset Flag "1" indicates that the current or most recent reset was activated by writing to the register RESET (0Fh). "0" indicates that register RESET has not been active. The first read of RSSTAT following the end of the firmware reset cycle clears URST to 0. URST is cleared to 0 by hardware reset.

HRST - bit 0 - Hardware Reset Flag indicates that the current or most recent reset was activated by hardware reset (pin RESET-). "0" indicates that pin RESET- has not been set to 0. The BSY flag is set whenever hardware reset is activated. The BSY flag and IDE interrupt are cleared by writing 1 followed by 0 to control bit CLRBSY in register HICTL (20h.4). The first read of RSSTAT following the end of the hardware reset cycle clears HRST to 0. HRST is not changed by firmware reset.

FIG 73 is a description of ATAPI Task File Registers (TR). The Task File register bits are labelled according to the ATAPI Specification.

ATFEA - Output from Features TR - The host writes this register at host address 1F1h.

ATERR - Input to Error TR - The host reads this register at host address 1F1h.

FIG. 74 is a description of ATINT - I/O of Interrupt Reason TR - The host accesses this register at host address 1F2h. (1F2h is Sector Count in ATA Specification.)

FIG. 75 is a description of ATSPA - Spare TR (unused in ATAPI Specification) - The host accesses this register at host address 1F3h. (1F3h is Sector Number in ATA Specification.)

FIG. 76 is a description of ATBLO - I/O of Byte Count Low TR - The host accesses this register at host address 1F4h. (1F4h is Cylinder Low in ATA Specification.)

FIG. 77 is a description of ATBHI - I/O of Byte Count High TR - The host accesses this register at host address 1F5h. (1F5h is Cylinder High in ATA Specification.)

FIG. 78 is a description of ATDRS - I/O of Drive Select TR - The host accesses this register at host address 1F6h. Bit 4, DRV, selects drive 1 when high or drive 0 when low. Bit 6, L, should be set to 1 to select LBA (not CHS) addressing. Bit 4 (DRV) is set to 1 by hardware reset. (1F6h was Drive/Head Select in ATA Specification.)

FIG. 79 is a description of ATCMD - Output from Command Register - The host writes this register at host address 1F7b.

ATSTA - Input to Status Register - The host reads this register at host address 1F7h.

FIGS. 80-87 are descriptions of the Microcontroller to Host Data Transfer Registers. The microcontroller writes up to eight bytes of daua to be transferred to UDTA0-7. The host reads these registers as data at host address 1F0h. UDTA0 is read first, and UDTA7 last. See the description of control bits UDTRG and UDATA in register HDDIR (1Fh.7,6).

Claims

1. A compact disk drive controller for a compact disk drive to control the communication of digital information between a compact disk and a host computer, said compact disk drive having drive electronics including a digital signal processor, a microcontroller, a random access memory and a system controller, wherein said host computer communicates with said compact disk drive controller via an IDE data bus and receives digital information from the compact disk via said IDE data bus. said compact disk drive controller comprising:

    a host interface connecting said host computer via said IDE data bus with said compact disk drive controller, to receive data addresses and commands from said host computer and to transmit digital information to said host computer;

    a path for communicating data addresses and commands from said host interface to the mi-

crocontroller of said drive electronics; and

    a digital signal processor interface connecting said host interface and said digital signal processor of said drive electronics to receive digital information from said compact disk and to transmit said digital information to said host interface.

2. The compact disk drive controller of claim 1, wherein said digital signal processor interface further comprises a descrambler to descramble and assemble the digital information received from said digital signal processor and store said digital information into said random access memory.

3. The compact disk drive controller of claim 1, wherein said digital signal processor interface further comprises an error correction code circuit to perform error correction on said digital information.

4. The compact disk drive controller of claim 3, wherein said error correction code circuit performs Reed-Solomon error correction on said digital information.

5. The compact disk drive controller of claim 3, wherein said digital signal processor interface further comprises a cyclic redundancy checker for detecting errors in said digital information after correction of said digital information by said error correction code circuit.

6. The compact disk drive controller of claim 1, wherein said host interface receives data addresses and commands from said host computer via said ISA data bus and communicates digital information to said host computer via said ISA bus.

7. The compact disk drive controller of claim 1, wherein said host interface further comprises a command FIFO to transfer commands from the host computer to the system controller of said drive electronics of said compact disk drive.

8. The compact disk drive controller of claim 1, wherein said host interface further comprises a status FIFO to transfer status information from said drive electronics of said compact disk drive to said host computer.

9. The compact disk drive controller of claim 1, wherein said drive electronics of said compact disk drive further comprises an ISA data bus and said host interface further comprises a configuration register via which said host computer in-

structs said compact disk drive controller to present said digital information onto one of said ISA and IDE data buses in a data format selected from a group including 16-bit DMA, 8-bit DMA, 16-bit PIO, and an 8-bit PIO format.

10. In a disk drive controller comprisiing a host interface and a digital signal processor interface, a method of controlling the communication of digital information between a host computer with an IDE data bus and a compact disk, comprising the steps of:

receiving commands from said host computer via said IDE data bus;

receiving data from a digital signal processor of a compact disk drive; and

communicatiing said data via said IDE data bus to said host computer.

11. The method of claim 10 wherein data received from said digital signal processor is corrected before being communicated to via said IDE data bus to said host computer.
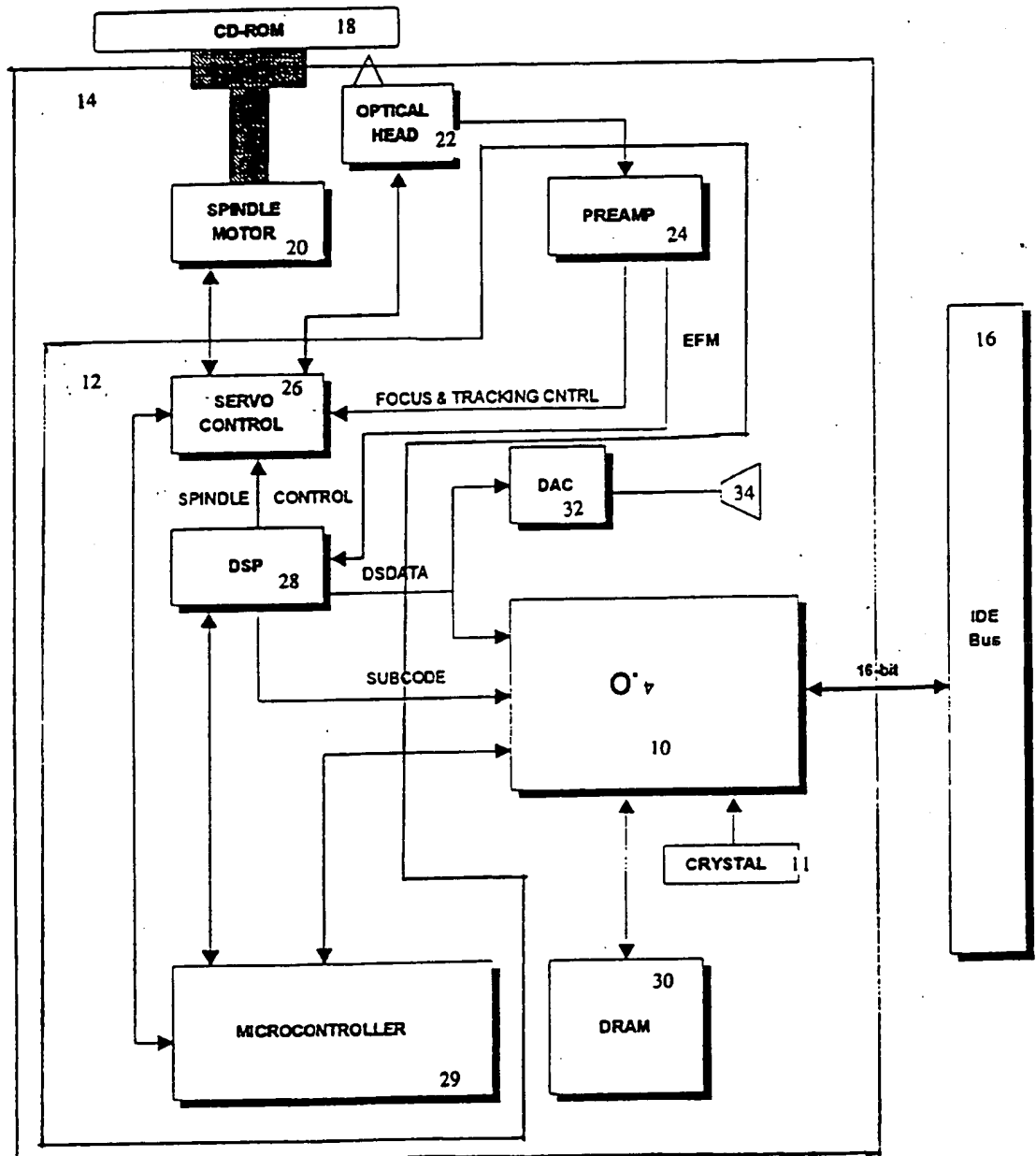
FIG. 1
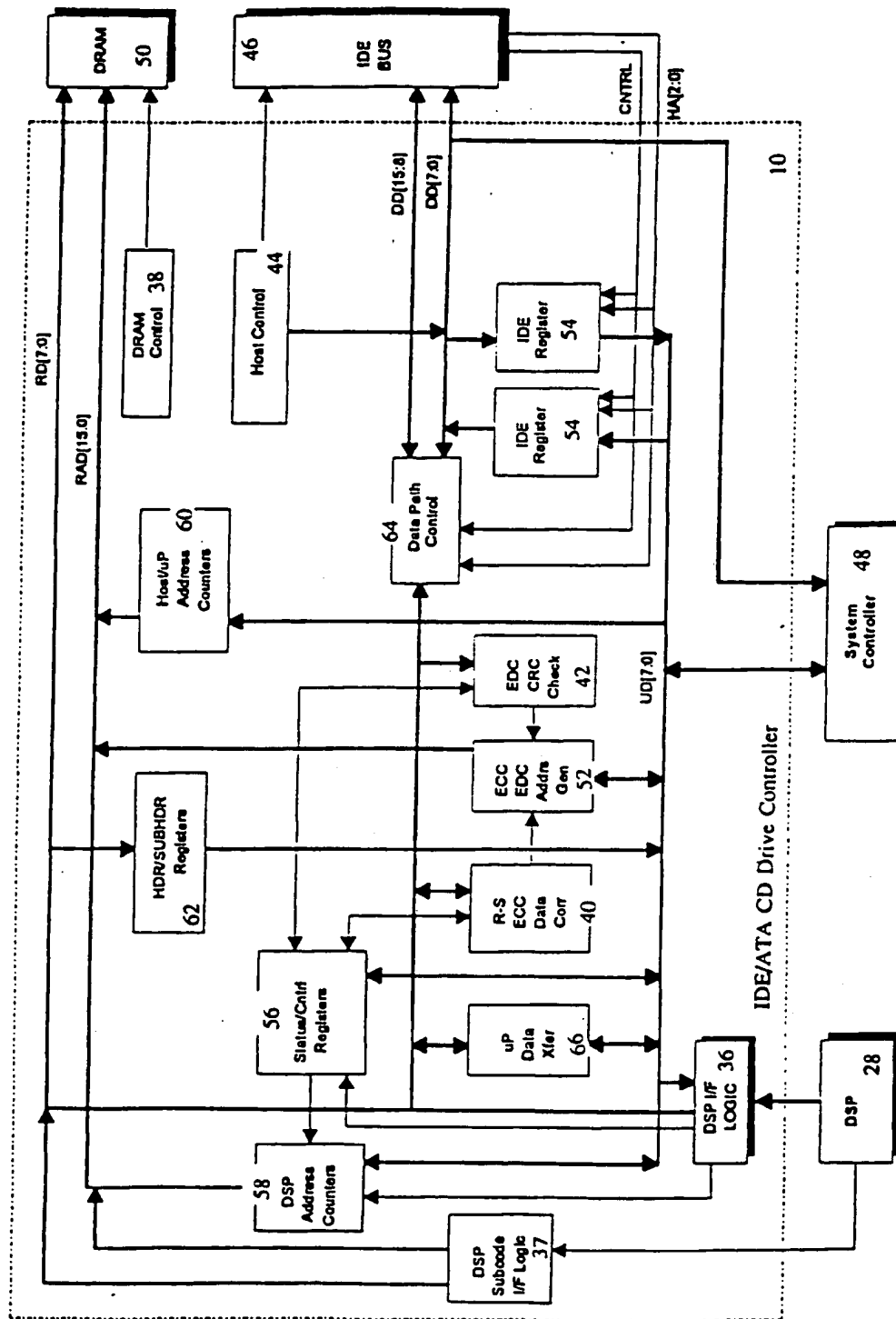
FIG. 2
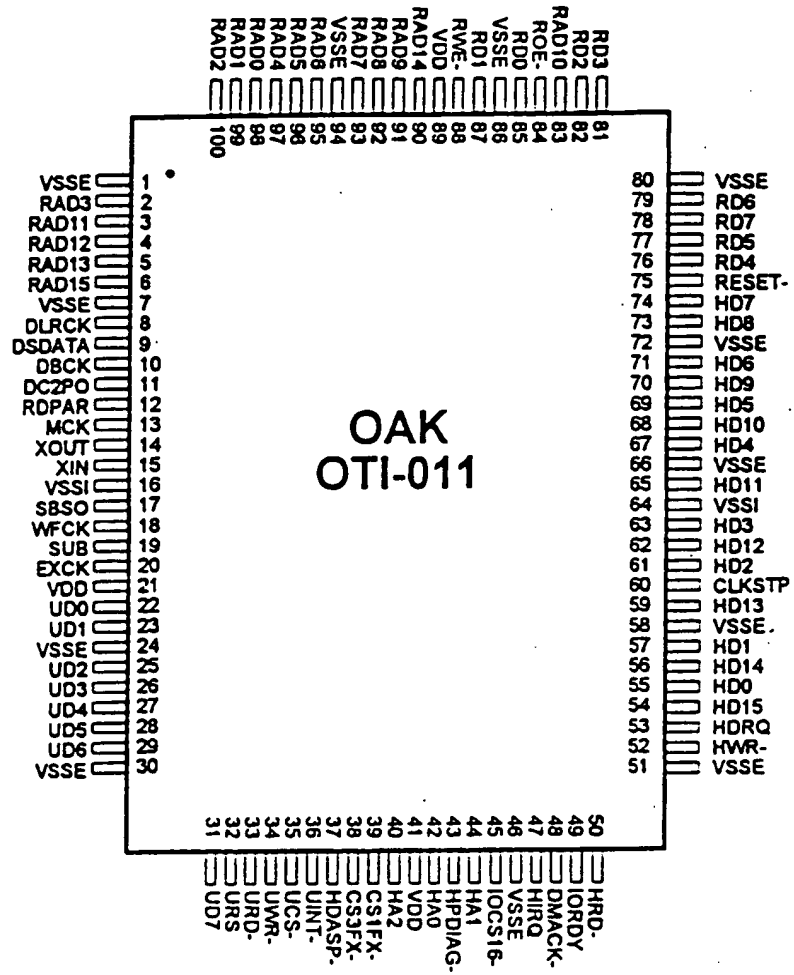
FIG. 3a

Pin #	Pin Name	Type	Comments	Pin #	Pin Name	Type	Comments
1	VSSE	P		29	UD6	B	uCOM Data Bus
2	RAD3	O	RAM Address Bus	30	VSSE	P	
3	RAD11	O		31	UD7	B	uCOM Data Bus
4	RAD12	O		32	URS	I	uCOM Register Select
5	RAD13	O		33	URD-	I	uCOM Read Strobe
6	RAD15	O		34	UWR-	I	uCOM Write Strobe
7	VSSE	P		35	UCS-	I	uCOM Chip Select
8	DLRCK	I	44.1KHz L/R Strobe	36	UINT-	O	uCOM Interrupt
9	DSDATA	I	Serial Data	37	HDASP-	B	IDE Host Interface
10	DBCK	I	Serial Bit Clock	38	CS3FX	I	
11	DC2PO	I	C2 Pointer Input	39	CS1FX	I	
12	RDPAR	B	RAM Parity Data	40	HA2	I	
13	MCK	O		41	VDD	P	
14	XOUT	O	X'tal OSC Output	42	HAO	I	IDE Host Interface
15	XIN	I	X'tal OSC Input	43	HPDIAG-	B	
16	VSSI	P		44	HA1	I	
17	SBSO	I	Subcode Interface	45	IOCS16-	O	
18	WFCK	I		46	VSSE	P	
19	SUB	I		47	HIRQ	O	IDE Host Interface
20	EXCK	B		48	DMACK-	I	
21	VDD	P		49	IORDY	O	
22	UDO	B	uCOM Data Bus	50	HRD-	I	
23	UD1	B		51	VSSE	P	
24	VSSE	P		52	HWR-	I	IDE Host Interface
25	UD2	B	uCOM Data Bus	53	HDRQ	O	
26	UD3	B		54	HD15	B	
27	UD4	B		55	HD0	B	IDE Host Data Bus
28	UD5	B		56	HD14	B	

FIG. 3b

Pin #	Pin Name	Type	Comments	Pin #	Pin Name	Type	Comments
57	HD1	B	IDE Host Data Bus	79	RD6	B	RAM Data Bus
58	VSSE	P		80	VSSE	P	
59	HD13	B	IDE Host Data Bus	81	RD3	B	RAM Data Bus
60	CLKSTP	I		82	RD2	B	
61	HD2	B	IDE Host Data Bus	83	RAD10	O	RAM Address Bus
62	HD12	B		84	ROE-	B	RAM Read Enable
63	HD3	B		85	RDO	B	RAM Data Bus
64	VSSI	P		86	VSSE	P	
65	HD11	B	IDE Host Data Bus	87	RD1	B	RAM Data Bus
66	VSSE	P		88	RWE-	O	RAM Write Enable
67	HD4	B	IDE Host Data Bus	89	VDD	P	
68	HD10	B		90	RAD14	O	RAM Address Bus
69	HD5	B		91	RAD9	O	
70	HD9	B		92	RAD8	O	
71	HD6	B		93	RAD7	O	
72	VSSE	P		94	VSSE	P	
73	HD8	B	IDE Host Data Bus	95	RAD6	O	RAM Address Bus
74	HD7	B		96	RAD5	O	
75	RESET-	I		97	RAD4	O	
76	RD4	B	RAM Data Bus	98	RAD0	O	
77	RD5	B		99	RAD1	O	
78	RD7	B		100	RAD2	O	

FIG. 3c

Pin Name	Pin Type	Pin Description
UD7-UD0	I/O	uCONTROLLER DATA BUS - Microcontroller bidirectional data bus. This bus is compatible with most microcontrollers.
URS-	IN	uCONTROLLER REGISTER SELECT - Register select input. AR (address register) is selected if URS = 0. Otherwise, microcontroller will access the internal register that is pointed by the AR register.
URD-	IN	uCONTROLLER READ STROBE - Microcontroller read strobe.
UWR-	IN	uCONTROLLER WRITE STROBE - Microcontroller write strobe.
UCS-	IN	uCONTROLLER CHIP SELECT - Microcontroller chip select input.
UINT-	OUT/ Open-Drain	uCONTROLLER INTERRUPT - Microcontroller interrupt output. This is an open-drain output. This signal can be externally wired - OR with other interrupt sources.

FIG. 4

Pin Name	Pin Type	Pin Description
HD15 - HD0	I/O	HOST DATA BUS - Host Bidirectional Data Bus.
HWR-	IN	HOST WRITE STROBE - Host IO write strobe.
HRD-	IN	HOST READ STROBE - Host IO read strobe.
HA2 - HA0	IN	HOST ADDRESS - These are the 3-bit Host Addresses asserted by the host to access a register or data port in the OTI-011.
IOCS16-	OUT / Open-Drain	DRIVE 16-bits I/O - For PIO transfer, IOCS16- indicates to the host system that the 16-bit data port has been addressed and OTI-011 is prepared to send or receive a 16-bit data word.
CS1FX-	IN	HOST CHIP SELECT 1 - This is the chip select signal decoded from the host address bus used to select the Command Block registers in the OTI-011.
CS3FX-	IN	HOST CHIP SELECT 3 - This is the chip select signal decoded from the host address bus used to select the Control Block registers in the OTI-011.
HDRQ	OUT	HOST DMA REQUEST - This signal will be asserted for DMA data transfer when OTI-011 is ready to transfer data to or from the host. This signal is used in a handshake manner with DMACK-. When a DMA operation is enabled, data transfers are 16-bits wide.
DMACK-	IN	HOST DMA ACKNOWLEDGE - This signal is used by the host in response to HDRQ.
HIRQ	OUT	HOST INTERRUPT - This signal is used to interrupt the host system.
IORDY	OUT	HOST IO READY - This signal is asserted to extend the host transfer cycle of any host read access when the OTI-011 is not ready to respond to a request.
HDASP-	I/O	HOST DRIVE ACTIVE/DRIVE 1 PRESENT - This is a time-multiplexed signal which indicates that a drive is active, or that Drive 1 is present. This signal is controlled by the firmware.
HPDIAG-	I/O	HOST PASSED DIAGNOSTICS - This signal is asserted by Drive 1 to indicate to Drive 0 that it has completed diagnostics. This signal is controlled by the firmware.

FIG. 5a

Addresses					Functions	
					Read (DIOR-)	Write (DIOW-)
CS1FX-	CS3FX-	DA2	DA1	DA0	Control block registers	
N	N	x	x	x	Data bus high impedance	Not used
N	A	1	1	0	Alternate ATAPI status	Device control
N	A	1	1	1	Drive address	Not used
					Command block registers	
A	N	0	0	0	Data	Data
A	N	0	0	1	ATAPI Error Register	ATAPI Features
A	N	0	1	0	ATAPI Interrupt Reason Register (R/W)	
A	N	0	1	1	Not Used	Not Used
A	N	1	0	0	ATAPI Byte Count Register (bit 7-0) (R/W)	
A	N	1	0	1	ATAPI Byte Count Register (bit 15-8) (R/W)	
A	N	1	1	0	Drive Select (R/W)	
A	N	1	1	1	ATAPI Status	ATA Command
A	A	x	x	· x	Invalid address	Invalid address

Logic conventions are: A = signal asserted, N = signal negated

FIG. 5b

Pin Name	Pin Type	Pin Description
DSDATA	IN	DSP SERIAL DATA - This serial input receives the data from DSP (CIRC decoder). The DBCK pin provides the data clock to the OTI-011. The received data are stored in the buffer RAM.
DBCK	IN	DSP BIT CLOCK - This clock input receives bit clock from DSP.
DLRCK	IN	DSP L/R CLOCK - Supplies the signal which is used to distinguish L-channel/R-channel (44.1KHz) DLRCK signal also indicates the start timing of 16 bit PCM word.
DC2PO	IN	DSP C2 POINTER - Supplies C2 error flag signal from DSP. When high, DC2PO indicates an error condition.

FIG. 6

Pin Name	Pin Type	Pin Description
SBSO	IN	SUBCODE SERIAL DATA - Supplies Subcode serial data. The subcode words (P-W) will be stored into buffer RAM in the order.
SUB	IN	SUBCODE SYNC - Supplies Subcodes SYNC from DSP.
EXCK	I/O	EXTERNAL CLOCK - Supplies Bit Clock for Subcode interface. This pin can be programmed as input or output.
WFCK	IN	WRITE FRAME CLOCK - Supplies Write Frame Clock from DSP. (7.35KHz)

FIG. 7

Pin Name	Pin Type	Pin Description
RAD15 - RAD0	OUT	RAM ADDRESS BUS - Buffer RAM address outputs.
RDPAR	I/O	RAM PARITY DATA - RAM Parity bit when RPEN = 1.
RD7 - RD0	I/O	RAM DATA BUS - Buffer RAM data bus.
RWE-	OUT	RAM WRITE ENABLE - The OTI-011 forces this pin low when writing data to the RAM.
ROE-	OUT	RAM OUTPUT ENABLE - The OTI-011 forces this pin low when reading data from the RAM.

FIG. 8a

Pin Name	Pin Function	Note
RA0 - RA8	DRAM address	
RA9		Left Open (see note)
RA10	CAS-	
RA11 - RA13		Left Open (see note)
RA14	RAS-	
RA15		Left Open (see note)

FIG. 8b

Miscellaneous Pins

Pin Name	Pin Type	Pin Description
XIN XOUT	IN OUT	XTAL INPUT / XTAL OUTPUT - XIN, XOUT are normally connected to a crystal. (up to 21MHz) XIN pin may be driven by the external signal. (21MHz max. 45 - 55% duty) The OTI-011 contains an internal resister between XIN and XOUT, so that an external resister shall not be connected to these pins.
MCK	OUT	M CLOCK - This output supplies clock signal of one-half the crystal frequency when register bit MCK1 is low. When register bit MCK1 is high, MCK pin supplies crystal frequency.
CLKSTP	IN	CLOCK STOP - CLKSTP stops the internal clock, when it is high.
RESET-	IN	RESET - Forcing this input low reset the OTI-011. OTI-011 can be reset by either this reset signal or the internal reset command (Writing RESET register). RSSTAT register shows user which reset most recently took place.
VDD		Supplies 5.0V+/- 5%
VSSE/VSSI		Ground Pins

FIG. 9

AR - Address Register									
Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
-	R	address							
-	W	address							

FIG. 10

COMIN - Command Packet Register									
Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
00h	R	binary							
00h	W	unused							

FIG. 11

IFSTAT/IFCTRL - Interface Status Register/Interface Control Register									
Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
01h	R	cmdib	dteib	decib	X	dtbsyb	X	dtenb	X
01h	W	cmdien	deteien	decien	0	0	0	douten	0

FIG. 12

DBCL - Data Byte/Word Counter									
Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
02h	R	b7	b6	b5	b4	b3	b2	b1	b0
02h	W	b7	b6	b5	b4	b3	b2	b1	b0

FIG. 13

DBCH - Data Byte/Word Counter									
Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
03h	R	dtei	dtei	dtei	dtei	b11	b10	b9	b8
03h	W	0	0	0	0	b11	b10	b9	b8

FIG. 14

HEAD0/DACL - Header Register 0/Data Address Counter									
Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
04h	R	header minutes (bcd)							
04h	W	a7	a6	a5	a4	a3	a2	a1	a0

FIG. 15

HEAD1/DACH - Header Registers/Data Address Counter									
Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
05h	R	header seconds (bcd)							
05h	W	a15	a14	a13	a12	a11	a10	a9	a8

FIG. 16

HEAD2/DTTRG - Header Registers/Data Transfer Trigger									
Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
06h	R	header frames (bcd)							
06h	W	(data unused)							

FIG. 17

HEAD3/DTACK - Header Registers/Data Transfer Acknowledge									
Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
07h	R	header mode (bcd)							
07h	W	(data unused)							

FIG. 18

STARTING POINT	MODE 1 (MODRQ=0)	MODE 2 (MODRQ=1)
First Sync Byte	FF,F0h	FF,E8h
First Header Byte	FF,FCh	FF,F4h
First Subheader Byte	-	FF,F8h
First User Data Byte	00,00h	00,00h

FIG. 19

PTL/WAL - ECC Block Pointer/Write Address Counter									
Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
08h	R	a7	a6	a5	a4	a3	a2	a1	0
08h	W	a7	a6	a5	a4	a3	a2	a1	a0

FIG. 20

PTH/WAH - ECC Block Pointer/Write Address Counter									
Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
09h	R	a15	a14	a13	a12	a11	a10	a9	a8
09h	W	a15	a14	a13	a12	a11	a10	a9	a8

FIG. 21

WAL/CTRL0 - Write Address Counter/Control 0 Register									
Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
0Ah	R	a7	a6	a5	a4	a3	a2	a1	0
0Ah	W	decen	0	e01rq	autorq	0	wrrq	qrq	prq

FIG. 22

DECEN 0Ah.7	WRRQ 0Ah.2	E01RQ 0Ah.5	QRQ 0Ah.1	PRQ 0Ah.0	DECODER OPERATION	CRCOK (0Ch.7)	HEAD0-3, SUBH0-3 (04-07h,14-17TH)
1	1	1	1	1	Q-P-correction	Valid	Buffer RAM
1	1	1	1	0	Q-correction	Valid	Buffer RAM
1	1	1	0	1	P-correction	Valid	Buffer RAM
1	1	X	0	0	Write-only (no ECC)	Valid	Buffer RAM
1	0	X	0	0	Disk-monitor	Invalid	Incoming data
1	0	X	X	1	Not recommended		(see note)
1	0	X	1	X	Not recommended		(see note)
0	X	X	X	X	Decoder disabled	Invalid	Invalid

FIG. 23

WAH/CTRL1 - Write Address Counter/Control 1 Register									
Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
0Bh	R	a15	a14	a13	a12	a11	a10	a9	a8
0Bh	W	syien	syden	dscren	cowren	modrq	formrq	mbckrq	shdren

FIG. 24

STAT0/PTL - Status 0 Register/ECC Block Pointer									
Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
0Ch	R	crcok	ilsync	nosync	lblk	wshort	sblk	0	uceblk
0Ch	W	a7	a6	a5	a4	a3	a2	a1	a0

FIG. 25

SYIEN 0Bh.7	SYDEN 0Bh.6	SYNC OPERATION	ILSYNC 0Ch.6	NOSYNC 0Ch.5	LBLK 0Ch.4	SBLK 0Ch.2
1	1	Sync-insertion +detection	active	active	inactive	inactive
1	0	Sync-insertion only	inactive	active	inactive	active
0	1	Sync-detection only	active	inactive	active	inactive

FIG. 26

RMO3 0Eh.7	RMOD2 0Eh.6	RMOD1 0Eh.5	RMOD0 0Eh.4	MODE
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	(8*N) + 0
1	0	0	1	(8*N) + 1
1	0	1	0	(8*N) + 2
1	0	1	1	(8*N) + 3
1	1	0	0	(8*N) + 4
1	1	0	1	(8*N) + 5
1	1	1	0	(8*N) + 6
1	1	1	1	(8*N) + 7 or error

FIG. 29

STAT1/PTH - Status 1 Register/ECC Block Pointer									
Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
0Dh	R	0	0	0	hdera	0	0	0	shdera
0Dh	W	a15	a14	a13	a12	a11	a10	a9	a8

FIG. 27

STAT2/SFCTRL									
Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
0Eh	R	rmod3	rmod2	rmod1	rmod0	mode	nocor	rform1	rform0
0Eh	W	unused							

FIG. 28

STAT3/RESET									
Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
0Fh	R	valstb	0	cblk	0	0	0	0	0
0Fh	W	(data unused)							

FIG. 30

CTRLW									
Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
10h	R	unused							
10h	W	0	swen	sdss	dclke	0	0	0	0

FIG. 31

CRTRG									
Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
11h	R	unused							
11h	W	0	0	0	0	0	0	0	crtrl

FIG. 32

Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
12h	R	unused							
12h	W	unused							

FIG. 33

Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
13h	R	unused							
13h	W	unused							

FIG. 34

SUBH0									
Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
14h	R	subheader file number							
14h	W	unused							

FIG. 35

SUBH1									
Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
15h	R	subheader channel number							
15h	W	unused							

FIG. 36

SUBH2									
Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
16h	R	subheader submode number							
16h	W	unused							

FIG. 37

SUBH3									
Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
17h	R	subheader coding information							
17h	W	unused							

FIG. 38

DECODER OPERATION	REGISTER	ERASURE BYTE #	STORED BYTE #
Disk-monitor	SUBH0	no erasures	incoming byte 16
Disk-monitor	SUBH0	byte 16	incoming byte 20
Disk-monitor	SUBH0	byte 16 and 20	incoming byte 20
Buffered	SUBH0	don't care	buffered byte 20
Disk-monitor	SUBH1	no erasures	incoming byte 17
Disk-monitor	SUBH1	byte 17	incoming byte 21
Disk-monitor	SUBH1	byte 17 and 21	incoming byte 21
Buffered	SUBH1	don't care	buffered byte 21
Disk-monitor	SUBH2	no erasures	incoming byte 18
Disk-monitor	SUBH2	byte 18	incoming byte 22
Disk-monitor	SUBH2	byte 18 and 22	incoming byte 22
Buffered	SUBH2	don't care	buffered byte 22
Disk-monitor	SUBH3	no erasures	incoming byte 19
Disk-monitor	SUBH3	byte 19	incoming byte 23
Disk-monitor	SUBH3	byte 19 and 23	incoming byte 23
Buffered	SUBH3	don't care	buffered byte 23

FIG. 39

VER/XTAL									
Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
1Ah	R	b7	b6	b5	b4	b3	b2	b1	b0
1Ah	W	0	0	0	0	mck1	0	0	xtald2

FIG. 40

DSPSL									
Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
1Bh	R				unused				
1Bh	W	c2ml	sel16o	1ch1	0	0	sel16	dir	edge

FIG. 41

UD7 c2ml	UD6 sell6o	UD5 1ch1	UD4 X	UD3 X	UD2 sell6	UD1 cir	UD0 edge	DSP Type
1	0	1	0	0	0	1	0	Sony CXD1135Q
1	1	0	0	0	0	1	1	Philips SAA7345
0	0	1	0	0	1	0	0	Sanyo LC7860K
0	0	0	0	0	1	1	1	Toshiba
0	0	1	0	0	1	0	1	Matsushita MN66251

FIG. 42

HCON/UACL									
Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
1Ch	R	unused							
1Ch	W	a7	a6	a5	a4	a3	a2	a1	a0

FIG. 43

DSPSL									
Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
1Bh	R	unused							
1Bh	W	c2ml	sel16o	1ch1	x	x	sel16	dir	edge

FIG. 44

HCON/UACL									
Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
1Ch	R	0	spare	spare	spare	dts	sdrqb	1ohi	dma16
1Ch	W	a7	a6	a5	a4	a3	a2	a1	a0

FIG. 45

UACH									
Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
1Dh	R	unused							
1Dh	W	a15	a14	a13	a12	a11	a10	a9	a8

FIG. 50

UACU									
Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
2Dh	W	0	0	0	0	b3	b2	b1	b0

FIG. 49

RAMRD/RAMWR									
Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
1Eh	R	rr7	rr6	rr5	rr4	rr3	rr2	rr1	rr0
1Eh	W	rw7	rw6	rw5	rw4	rw3	rw2	rw1	rw0

FIG. 51

HDDIR									
Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
1Fh	R	urtbsy	cs13	0	0	0	0	hrwr	0
1Fh	W	udtrg	udata	host16	0	1	1	0	0

FIG. 52

Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
0	1	1	0	0

FIG. 53

HICTL									
Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
20h	W	0	pdiagen	daspen	clrbsy	setbsy	scod	iordyen	iocs16en

FIG. 54

SUBC2									
Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
21h	W	0	0	0	0	nopq	cdsp2	cdsp1	cdsp0

FIG. 55

CDSP2 21h.2	CDSP1 21h.1	CDSP0 21h.0	Subcode Block Rate
0	0	0	1X (75 blocks/sec)
0	0	1	2X (150 blocks/sec)
0	1	0	4X (300 blocks/sec)
1	0	0	Reserved

FIG. 56

STATS									
Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
22h	R	data unused					missy	sbkend	silsy
22h	W	data unused							

FIG. 57

DBACL									
Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
24h	W	b7	b6	b5	b4	b3	b2	b1	b0

FIG. 59

DBACH									
Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
25h	W	0	0	0	0	0	0	0	b8

FIG. 60

SBKL									
Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
26h	RW	b7	b6	b5	b4	b3	b2	b1	b0

FIG. 61

SBKH									
Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
27h	RW	data unused							b8

FIG. 62

WBKL									
Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
28h	RW	b7	b6	b5	b4	b3	b2	b1	b0

FIG. 63

WBKH									
Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
29h	RW	data unused							b8

FIG. 64

RAMCF									
Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
2Ah	R	rftyp	ramclr	ram0	uhilo	rpen	rcf2	rcf1	rcf0
2Ah	W	rftyp	ramclr	0	uhilo	rpen	rcf2	rcf1	rcf0

FIG. 65

RCF2 2Ah.2	RCF1 2Ah.1	RCF0 2Ah.0	RAM Configuration
0	0	1	128K DRAM (256K X 4-bit X 1)
0	1	0	256K DRAM (256K X 4-bit X 2)

FIG. 66

MEMCF - Memory Layout Configuration Register									
Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
2Bh	W	0	0	0	0	purg	iordyf	mly1	mly0

FIG. 67

MLY1 2Bh.1	MLY0 2BH.0	DRAM Memory Layout Configuration
0	0	Data Blocks + Last 2 Auxiliary/Subcode Blocks
1	1	Data Blocks + All Auxiliary/Subcode Blocks

FIG. 68

SUBCD - Subcode Control									
Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
2Ch	W	sbxck	scen	sckb2	scien	exinv	exop	sbsel1	sbsel0

FIG. 69

EP 0 689 207 A1

SBSEL1 2Ch. 1	SBSEL0 2Ch.0	Subcode Format
0	0	SMD0 (Philips)
0	1	SMD1 (EIAJ-1)
1	0	SMD2 (EIAJ-2)
1	1	Undefined

FIG. 70

UMISC - Miscellaneous Microcontroller Control									
Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
2Eh	R	1	1	1	1	1	1	pdiagb	daspb
2Eh	W	ideien	0	drveb	drv1b	hintrq	0	0	0

FIG. 71

RSSTAT - Reset, IDE, and Subcode Status Register									
Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
2Fh	R	srst	cmd	diagcmd	sint	pannt	rst	urst	hrst

FIG. 72

ATFEA/ATERR									
Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
31h	R	0	0	0	0	0	0	0	dma
31h	W	b7	b6	b5	b4	mcr	abrt	eom	ili

FIG. 73

ATINT									
Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
32h	RW	0	0	0	0	0	0	io	cod

FIG. 74

ATSPA									
Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
33h	RW	0	0	0	0	0	0	0	1

FIG. 75

ATBLO									
Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
34h	RW	b7	b6	b5	b4	b3	b2	b1	b0

FIG. 76

ATBHI									
Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
35h	RW	b7	b6	b5	b4	b3	b2	b1	b0

FIG. 77

ATDRS									
Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
36h	RW	1	1	1	drv	0	0	0	0

FIG. 78

ATCMD/ATSTA									
Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
37h	R	b7	b6	b5	b4	b3	b2	b1	b0
37h	W	0	drdy	0	drc	drq	corr	0	check

FIG. 79

DFIF0									
Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
40h	W	b7	b6	b5	b4	b3	b2	b1	b0

FIG. 80

DFIF1									
Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
41h	W	b7	b6	b5	b4	b3	b2	b1	b0

FIG. 81

DFIF2									
Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
42h	W	b7	b6	b5	b4	b3	b2	b1	b0

FIG. 82

DFIF3									
Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
43h	W	b7	b6	b5	b4	b3	b2	b1	b0

FIG. 83

DFIF4									
Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
44h	W	b7	b6	b5	b4	b3	b2	b1	b0

FIG. 84

DFIF5									
Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
45h	W	b7	b6	b5	b4	b3	b2	b1	b0

FIG. 85

DFIF6									
Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
46h	W	b7	b6	b5	b4	b3	b2	b1	b0

FIG. 86

DFIF7									
Address	Type	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
47h	W	b7	b6	b5	b4	b3	b2	b1	b0

FIG. 87

European Patent
Office

EUROPEAN SEARCH REPORT

Application Number

EP 95 20 1658

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.6)
Y	WO-A-93 09500 (TANDY COTPORATION) * figure 1 * * page 7, line 11 - page 8, line 18 * ---	1-11	G11B27/00 G06F3/06
Y	US-A-5 295 247 (CHANG ET AL.,) * abstract * * figure 1 * ---	6,9	
Y	ELECTRONIC DESIGN, vol.40, no.20, 1 October 1992, CLEVELAND, OHIO pages 79 - 85 RICHARD NASS 'DISK-DRIVECONTROLLERS AUTOMATE DATA FLOW' -----	1-5,7,8, 10,11	

TECHNICAL FIELDS
SEARCHED   (Int.Cl.6)

G06F

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	13 October 1995	Weiss, P